UNIVERSITY OF
ILLINOIS LIBRARY
AT URBANA-CHAMPAIGN
ENGINEERING

JUN 20 1980

The person charging this material is responsible for its return to the library from which it was withdrawn on or before the **Latest Date** stamped below.

Theft, mutilation, and underlining of books are reasons for disciplinary action and may result in dismissal from the University.

To renew call Telephone Center, 333-8400

UNIVERSITY OF ILLINOIS LIBRARY AT URBANA-CHAMPAIGN

ENGINEERING

PHOTO REPRODUCTION

JUN - 4 REC'D

INTERLIBRARY LOAN

JUN 19 REC'D

L161—O-1096

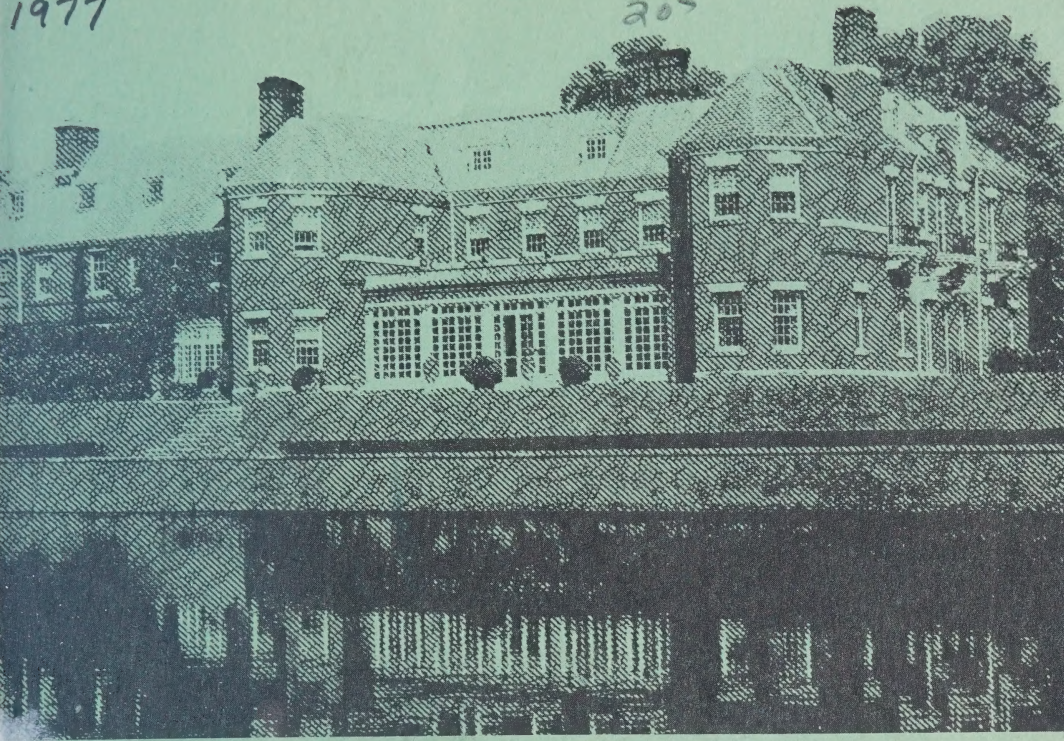
Digitized by the Internet Archive
in 2026 with funding from
University of Illinois Urbana-Champaign

https://archive.org/details/proceedings1977alle_0

621.3811
Al5
1977

engin.

20



Proceedings

FIFTEENTH ANNUAL ALLERTON CONFERENCE ON COMMUNICATION, CONTROL, AND COMPUTING

September 28-30, 1977

Allerton House, Monticello, Illinois
Sponsored by the
Department of Electrical Engineering and the
Coordinated Science Laboratory of the
University of Illinois at Urbana-Champaign

The Library of the
APR 29 1980
University of Illinois
at Urbana-Champaign

ENGINEERING LIBRARY

ENGINEERING LIBRARY
UNIVERSITY OF ILLINOIS
URBANA, ILLINOIS

PROCEEDINGS
FIFTEENTH ANNUAL ALLERTON CONFERENCE ON COMMUNICATION, CONTROL, AND COMPUTING

F. P. Preparata
M. B. Pursley
Conference Co-Chairmen

Conference held

September 28-30, 1977
Allerton House
Monticello, Illinois

Sponsored by

The Department of Electrical Engineering
and
The Coordinated Science Laboratory
of the
UNIVERSITY OF ILLINOIS
at
Urbana-Champaign

THE LIBRARY OF THE
JAN 23 1978
UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN

621.3811
Al5
1977

ENGINEERING LIBRARY

FOREWORD

The annual Allerton Conference, which was held for the past fourteen years under the title "Allerton Conference on Circuit and System Theory," was held this year under a new title which reflects the evolving interests of its audience. The Fifteenth Annual Allerton Conference on Communication, Control, and Computing was held at the Allerton House on September 28-30, 1977.

The Conference was sponsored by the Department of Electrical Engineering and the Coordinated Science Laboratory of the University of Illinois at Urbana-Champaign. These PROCEEDINGS contain the papers presented at the Conference.

In addition to the large number of submitted papers, there were several organized sessions of invited papers. These were four sessions on Computational Complexity organized by C. L. Liu and F. P. Preparata, two sessions on Stochastic Models in Engineering Systems organized by G. Blankenship, and one session each on Communication Theory, organized by K. Yao, and Multi-Criteria Decision Making, organized by J. D. Medanić. We are grateful to these organizers for their time and effort on behalf of the Conference. We are also grateful to the other session chairmen and authors for their contributions to the success of the Allerton Conference.

On Wednesday evening R. L. Rivest delivered the plenary address entitled "On Digital Signatures and Public-Key Crypto-Systems." This meeting was co-sponsored by the Joint Chapter of the IEEE Information Theory Group and was chaired by its chairman M. B. Pursley.

The support and encouragement of Professor E. C. Jordan, Head of the Department of Electrical Engineering and Professor R. T. Chien, Director of the Coordinated Science Laboratory are appreciated and acknowledged. We thank our colleagues for their generous assistance in organizing the Conference: J. Abraham, D. Bertsekas, J. B. Cruz, Jr., E. S. Davidson, R. A. Flower, A. H. Haddad, W. K. Jenkins, P. V. Kokotović, J. Medanić, G. Metze, W. R. Perkins, H. V. Poor, D. V. Sarwate, T. N. Trick, and M. E. Van Valkenburg.

We are grateful to Harold Lawler for his handling of the business aspects of the Conference. Thanks are also due to J. P. Rutledge for coordinating the projectionists and the visual aid equipment. Special thanks are due to Rosa Townsend and Phyllis Young for their expert and skillful coordination of the Conference arrangements. Also we want to thank Phyllis Young for her help in the preparation and editing of the Conference PROCEEDINGS.

F. P. Preparata
M. B. Pursley

TABLE OF CONTENTS

FOREWORD iii

I-A: COMPUTATIONAL COMPLEXITY I

Co-Organizers: C. L. Liu and F. P. Preparata
University of Illinois

Chairman: G. K. Manacher
University of Illinois, Chicago

EFFICIENT SPECIAL PURPOSE PRIORITY QUEUES 1
D. B. Johnson

POLYNOMIAL b-BALANCED TREES 8
F. Luccio and L. Pagli

DATA ARRANGEMENT IN MAGNETIC BUBBLE MEMORIES 16
C. K. Wong

FAST SIMULATION OF COMBINATIONAL LOGIC NETWORKS BY MACHINES WITHOUT RANDOM-ACCESS STORAGE 25
Nicholar Pippenger

THE COMPLEXITY OF COORDINATING PARALLEL ASYNCHRONOUS PROCESSES 34
H. T. Kung

I-B: STOCHASTIC MODELS IN ENGINEERING SYSTEMS I

Chairman and Organizer: G. Blankenship
Case Western Reserve
University and ERDA

PROBABILITIES OF LARGE DERIVATIONS FOR SOME STOCHASTIC EQUATIONS AND APPLICATIONS 44
G. C. Papanicolaou

OPTIMAL FINITE DIMENSIONAL RECURSIVE ESTIMATORS FOR DISCRETE-TIME STOCHASTIC NONLINEAR SYSTEMS 52
S. I. Marcus

QUANTUM DETECTION AND ESTIMATION THEORY 62
S. K. Mitter and S. K. Young

CONTINUOUS QUANTUM FILTERING 68
John S. Baras

I-C: CONTROL THEORY I

Chairman: J. B. Cruz, Jr.
University of Illinois

ON MODEL REFERENCE ADAPTIVE CONTROL SYSTEMS 78
R. V. Monopoli

MULTIPLE TIME SCALES IN SINGLE INPUT-SINGLE OUTPUT HIGH GAIN FEEDBACK SYSTEMS 85
Kar-Keung D. Young

STATE VARIABLE RECONSTRUCTION OF LINEAR DISCRETE SYSTEMS FROM DELAYED OBSERVATIONS 94
Dong H. Chyung

ZERO-SUM STRATEGY FOR SYSTEMS WITH FAST AND SLOW MODES 96
B. F. Gardner, Jr.

TRANSFORMATION OF LINEAR MULTIVARIABLE SYSTEMS TO CANONICAL FORMS 104
J. Hickin and N. K. Sinha

THEORY OF CONE METHODS FOR NULL-CONTROLLABILITY OF NONLINEAR CONTROL SYSTEMS 114
A. S. C. Sinha

I-D: DIGITAL SYSTEMS AND ARITHMETIC

Chairman: P. Maestrini
University of Pisa

THE GENERATION OF PROGRAMMED LOGIC ARRAY (PLA) IMPLEMENTATIONS FROM HIGHER LEVEL LANGUAGE DESCRIPTIONS 120
J. W. Gault

DIGITAL SYSTEM DESIGH USING 4-VALUED LOGIC DEVICES 128
A. S. Wojcik

AN ALGEBRA FOR THE REALIZATION OF SWITCHING FUNCTIONS USING A CERTAIN TYPE OF MOS PACKAGE 136
R. Nair and G. Metze

A MULTICODE SINGLE TRANSITION TIME STATE ASSIGNMENT FOR ASYNCHRONOUS SEQUENTIAL MACHINES 146
J. G. Kuhl and S. M. Reddy

QUANTIZATION ERROR IN FLOATING-POINT ARITHMETIC 156
A. B. Sripad and D. L. Snyder

FRACTIONAL RADIX NUMBER SYSTEMS AND THEIR APPLICABILITY TO AUTONOMOUS ONE-DIMENSIONAL DISCRETE-TIME SYSTEMS 161
W. L. Klein and M. E. Kaliski

LIST SEARCHES USING MULTIPLE PROCESSORS 162
L. D. Coraor and J. P. Robinson

PROPERTIES OF A MODIFIED INCIDENCE MATRIX FOR COMPUTATION GRAPHS 163
R. W. Church and T. Murata

II-A: COMPUTATIONAL COMPLEXITY II

Co-Organizers: C. L. Liu and F. P. Preparata
University of Illinois

Chairman: F. P. Preparata

BINARY SEARCH USING UNRELIABLE COMPARISONS 165
R. L. Rivest, A. R. Meyer,
D. J. Kleitman and J. Spencer

ALGORITHM DESIGN TECHNIQUES 175
Sartaj Sahni

COMBINATORIAL INFERENCE 186
R. DeMillo, D. Dobkin, and R. Lipton

A PROBLEM IN MULTIVARIATE STATISTICS: ALGORITHM,
DATA STRUCTURE AND APPLICATIONS 193
J. L. Bentley and M. I. Shamos

COUNTER MACHINES AND NUMBER-THEORETIC PROBLEMS 202
O. H. Ibarra

II-B: ESTIMATION AND IDENTIFICATION

Chairman: J. S. Baras
University of Maryland

ON THE ESTIMATION OF DISCRETE PROBABILITY
DENSITIES FROM NOISY MEASUREMENTS 211
L. P. Devroye and G. L. Wise

ROBUST IDENTIFICATION OF PARAMETERS OF
LINEAR SYSTEMS 221
S. Basu and V. D. VandeLinde

A NONPARAMETRIC STOCHASTIC APPROXIMATION ALGORITHM
FOR ON-LINE IDENTIFICATION OF MULTIVARIABLE SYSTEMS 231
H. El-Sherief and N. K. Sinha

ON OPTIMUM LINEAR ESTIMATION OF DIGITAL SEQUENCES 237
E. D. Mese and D. Giuli

II-C: DIGITAL COMMUNICATIONS AND DATA COMPRESSION

Chairman: H. V. Poor
University of Illinois

A VOLTERRA SERIES APPROACH TO THE TIME JITTER
IN A CHAIN OF REGENERATIVE REPEATERS 244
G. L. Cariolaro and F. C. Vagliani

NOISELESS OVERFULL TREE-ENCODING OF MEMORYLESS
SKEWED BINARY SOURCES 254
L. Th. N. M. Mulder and D. L. Cohn

COMPARISON OF SPC, PCM, AND ADPCM 264
N. C. Gallagher

DIGITAL FILTERING OF DELTA MODULATION ENCODED SIGNALS 271
D. P. Prezas and J. L. LoCicero

TRACKING PROPERTIES OF PHASE-LOCKED LOOPS IN OPTICAL RECEIVERS 280
U. Mengali and E. Pezzani

ON A MULTILEVEL QUANTIZED DIGITAL PHASE-LOCKED LOOP 289
F. Russo and N. A. D'Andrea

II-D: CONTROL THEORY II

Chairman: R. Monopoli
University of Massachusetts

AN ALGORITHM FOR THE SIMULATION OF LARGE SCALE SYSTEMS DESCRIBED BY MIXED ALGEBRAIC AND DIFFERENTIAL EQUATIONS 298
N. K. Tripathi and E. J. Davison

EIGENVALUE SENSITIVITY OF AGGREGATED MODELS 305
N. K. Sinha and Z. Elrazaz

THE NESTED BORDERED-DIAGONAL METHOD FOR SYMMETRIC EIGENPROBLEMS 314
P. P. Groumpos and P. D. Scott

NORMALIZATION OF SYSTEM VARIABLES AND MULTIVARIABLE SYSTEM ANALYSIS 324
R. Plackovic

GRADIENT ESTIMATION ALGORITHMS FOR ADAPTIVE EQUILIZATION IN THE DISCRETE FREQUENCY DOMAIN 337
P. Corsini, G. Picchi and G. Prati

ON A CLASS OF POLYTOPE INTERSECTION PROBLEMS 347
B. R. Barmish and J. Sankaran

III-A: COMPUTATIONAL COMPLEXITY III

Co-Organizers: C. L. Liu and F. P. Preparata
University of Illinois

Chairman: C. L. Liu

SPACE-TIME TRADEOFFS ON THE FFT ALGORITHM 359
J. E. Savage and S. Swamy

THE PROBABILISTIC ANALYSIS OF MATCHING HEURISTICS 368
C. H. Papadimitriou

APPROXIMATE SOLUTIONS FOR THE TWO-PARTITION PROBLEM 379
C. E. Kim

AN INFORMATION STRUCTURE FOR POSETS 389
E. Lodi and F. Luccio

THE FORD-JOHNSON SORTING ALGORITHM IS NOT OPTIMAL 390
G. K. Manacher

ON THE AVERAGE NUMBER OF MAXIMA IN A SET OF VECTORS AND APPLICATIONS 398
J. L. Bentley, H. T. Kung, M. Schkolnick and C. D. Thompson

III-B: RANDOM PROCESSES AND APPLICATIONS

Chairman: J. B. Thomas
Princeton University

INFORMATION-SINGULARITY AND RECOVERABILITY OF RANDOM PROCESSES 406
B. Hajek

LINEAR DIMENSIONALITY REDUCTION OF STATIONARY VECTOR GAUSSIAN PROCESSES 415
P. Papantoni-Kazakos, R. Li, and D. Kazakos

ON REDUCING MULTIPLICATIONS IN AN EXTENDED GENERALIZED WIENER FILTER 421
K. Yao

ROBUST ESTIMATION USING THE ROBBINS-MONRO STOCHASTIC APPROXIMATION ALGORITHM 429
E. L. Price and V. D. VandeLinde

ON ASYMPTOTICALLY PERFECT DISCRIMINATION WITH QUANTIZED DATA 438
D. Kazakos

III-C: CODING FOR ERROR CONTROL

Chairman: D. V. Sarwate
University of Illinois

SOME RESULTS ON BINARY CONVOLUTIONAL CODES FOR MULTIPLE ACCESS CHANNEL 439
R. Peterson and D. J. Costello, Jr.

A QUADRATIC CONVOLUTIONAL CODE 448
J. P. Robinson and D. S. Raju

GENERALIZED MINIMUM DISTANCE DECODING OF COMPLETELY ORTHOGONALIZABLE BINARY CODES 449
C. C. Yu and D. J. Costello, Jr.

APPLICATION OF THE R_o CRITERION TO SPREAD SPECTRUM COMMUNICATIONS 450
H. de Pedro and A. Giordano

ERROR PERFORMANCE OF REDUCED-STATE VITERBI DECODING 451
J. Chammas, D. Haccoun and J. Conan

TIME VARYING CONVOLUTIONAL CODES - A SYSTEM THEORY APPROACH 452
J. S. Bedi, H. Singh, A. K. Kamal

III-D: CONTROL THEORY III

Chairman: W. B. Rouse
University of Illinois

GOAL ANALYSIS: IMPLEMENTATION OF CONTROL ACTIONS USING DECISION TABLES 460
R. F. Erlandson

A MODEL OF HUMAN DECISION MAKING IN A FAULT DIAGNOSIS TASK 466
W. B. Rouse

MULTIPLE TIME SCALES FOR POWER SYSTEM ANALYSIS 476
J. J. Allemong and J. H. Chow

CLOSED LOOP COORDINATED CONTROL OF LARGE POWER PLANTS 486
N. Gopalsami and C. K. Sanathanan

ENERGY DECAY ESTIMATES AND CONTROL THEORY FOR THE WAVE EQUATION IN A BOUNDED DOMAIN 496
G. Chen

IV-A: COMMUNICATION THEORY

Chairman and Organizer: K. Yao
University of California,
Los Angeles

SOME PROBLEMS IN COMMUNICATION NETWORKS 503
P. Papantoni-Kazakos

ROBUST DETECTION OF STOCHASTIC SIGNALS 507
H. V. Poor and J. B. Thomas

TWO-INPUT CORRELATOR DETECTORS FOR RANDOM SIGNALS 517
J. G. Shin and S. A. Kassam

THE PERFORMANCE AND SENSITIVITY OF TWO FINITE-MEMORY DETECTORS 527
M. D. Wood and J. B. Thomas

ON UPPER AND LOWER BOUNDS OF ERROR PROBABILITY OF A FINITE-PRECISION DIGITALLY IMPLEMENTED RECEIVER 537
S. Reisenfeld and K. Yao

IV-B: COMPUTATIONAL COMPLEXITY IV

Co-Organizers: C. L. Liu and F. P. Preparata
University of Illinois

Chairman: D. E. Muller
University of Illinois

AN ALGORITHM FOR FINDING HAMILTONIAN CIRCUITS IN CERTAIN GRAPHS 547
R. E. Bixby and D. L. Wang

FAST ALGORITHMS FOR FINDING MINIMUM SPANNING TREES IN K DIMENSIONS 553
A. C. C. Yao

FINDING SMALLEST SPANNING FORESTS AND TREES SUBJECT TO MANY DEGREE CONSTRAINTS 557
H. N. Gabow

COMPUTATIONAL COMPLEXITY OF k-MAXIMAL CUTS 567
T. Nishizeki, H. Ogawa and N. Saito

HEURISTIC ALGORITHMS FOR THE SET COVERING PROBLEM AND THEIR BEHAVIOR 577
K. Y. Chwa and S. L. Hakimi

IV-C: MULTI-CRITERIA DECISION MAKING

Chairman and Organizer: J. D. Medanić
University of Illinois

MULTICRITERIA OPTIMIZATION WITH UNCERTAINTY IN THE DYNAMICS 587
W. E. Schmitendorf

EQUILIBRIUM CONCEPTS IN STOCHASTIC MULTICRITERIA OPTIMIZATION 595
D. A. Castanon

MACRO-ECONOMIC STABILIZATION POLICY FOR AN UNCERTAIN DYNAMIC ECONOMY 604
G. Leitmann and H. Y. Wan, Jr.

ON THE CONCEPTUAL FOUNDATIONS OF NORMATIVE "PUBLIC POLICY ANALYSIS": AN INTERDISCIPLINARY INTERPRETATION AND SYNTHESIS 613
H. W. Brock

IV-D: CIRCUIT THEORY

Chairman: M. E. Van Valkenburg
University of Illinois

BY-PASSING THE FREQUENCY DOMAIN 623
D. Hazony

SINGULAR POLE ANALYSIS OF ACTIVE FILTERS 632
A. M. Davis and E. A. Moustakas

MODIFICATION OF LUMPED LINEAR TIME-INVARIANT SYSTEMS BY CONTROLLED SOURCES FOR TRANSFER FUNCTION MATRIX SYNTHESIS 640
R. Perry and J. D'Alessandro

TIME LIMITED RLC NETWORKS 642
D. Hazony and Y. Hazony

V-A: STOCHASTIC MODELS IN ENGINEERING SYSTEMS II

Chairman and Organizer: G. Blankenship
Case Western Reserve
University and ERDA

STABILITY RESULTS FOR A CLASS OF SYSTEMS WITH MULTIPLICATIVE STATE NOISE 652
G. L. Wise and S. I. Marcus

GEOMETRY OF PARAMETRIC MODELS: SOME PROBABILISTIC QUESTIONS 661
P. S. Krishnaprasad

A REPRESENTATION THEOREM FOR LINEAR DIFFERENTIAL EQUATIONS WITH MARKOVIAN COEFFICIENTS 671
R. W. Brockett and G. L. Blankenship

ON STOCHASTIC OPTIMAL CONTROL WITH AN EXPONENTIAL-OF-INTEGRAL COST CRITERION 680
P. R. Kumar and J. H. Van Schuppen

V-B: COMPUTER-COMMUNICATION NETWORKS

Chairman: J. A. Abraham
University of Illinois

ON THE STATE-SPACE APPROACH IN MODELING DATA-COMMUNICATION NETWORKS 690
J. S. Meditch

ROUTING TO MULTIPLE SINKS IN A COMMUNICATION NETWORK 700
C. A. Wolfe, C. L. Morefield,
H. J. Payne, and B. Gardner

ON THE STATISTICAL ANALYSIS OF CONCENTRATORS WITH ARQ RETRANSMISSION SCHEMES 709
D. Towsley and J. K. Wolf

V-C: FAULT-TOLERANT DIGITAL SYSTEMS

Chairman: G. Metze
University of Illinois

REFERENCELESS SIGNATURE TESTING 718
J. G. Quam and S. M. Reddy

FAULT DIAGNOSIS OF MODULAR NETWORKS WITH A SMALL NUMBER OF FAULTS 727
G. G. L. Meyer

REDUNDANCY RESULTS IN COMBINATIONAL NETWORKS BY MEANS OF L-EXPRESSIONS 732
V. K. Agarwal and G. M. Masson

SELF-CHECKING ALTERNATING LOGIC:
COMBINATIONAL NETWORK ANALYSIS 742
S. E. Woodard and G. Metze

THE DESIGN OF TOTALLY SELF-CHECKING
ARITHMETIC CIRCUITS 751
J. Dussault and G. Metze

EASILY TESTABLE BINARY CIRCUITS CONSTRUCTED
WITH MULTIVALUED MORPHIC GATES 760
J. Dussault

EFFICIENT SPECIAL PURPOSE PRIORITY QUEUES

DONALD B. JOHNSON
Computer Science Department
The Pennsylvania State University
University Park, Pennsylvania 16802

ABSTRACT

Priority queues, as usually implemented, require $O(\log n)$ time for each INSERT and MIN operation when a sequence containing n INSERT operations is executed. It is shown how to reduce the average operation cost to $o(\log n)$ when there is a large number of UPDATE operations and to $o(\log s)$ for certain sequences when the domain of integer priorities is of size s. Data structures and application to routing and connection problems are discussed.

PRIORITY QUEUES

A priority queue with operations INSERT and MIN can be implemented so each operation will require $O(\log n)$ time, where there are at most n elements in the queue.† A standard method among the numerous methods which have been proposed [1,2,3,8,9] stores elements in a heap, a binary tree in which each path from the root is nondecreasing in priority. Heap order can be maintained under both INSERT and MIN by adjustments which are confined to the vicinity of a path in the tree of length $\lfloor \log n \rfloor$. If the domain of priorities is an arbitrary set with a total order, a running time of $o(\log n)$ cannot be achieved simultaneously for both operations since such a result would give an algorithm for comparison sorting which runs in $o(n \log n)$ time. Tradeoffs are easily obtained, however, in which either INSERT or MIN requires $O(n)$ time. If the queue is stored as an unordered list, then INSERT takes constant time and MIN requires $O(n)$ time. If the queue is stored as a sorted list then MIN takes constant time while INSERT is $O(n)$. More refined tradeoffs can be had with a repetoire of operations which includes UPDATE, the change of the priority of a queue element for the better [7]. The number of UPDATEs may dominate the other operations in certain algorithms with embedded priority queues, for example, algorithms for finding optimum spanning trees [7] and shortest paths [6]. For such problems a priority queue in which UPDATE and INSERT are asymptotically less costly than MIN can be desirable. We call such a priority queue an unbalanced priority queue. In contrast, the conventional queues in which INSERT and MIN are of equal asymptotic cost are called balanced.

It may be that the domain of priorities is restricted to some set of integers. In the case where the priorities are drawn from the set $\{1,\ldots,s\}$ the queue operations mentioned can be implemented to run in $O(\log\log s)$ time [10,11]. This result is achieved on a RAM under the uniform cost criterion (see [1]). The rather intricate data structure used is initialized at a cost of $O(s)$ which, when distributed over a sequence of $O(s)$ queue operations, does not dominate the sum of the individual operation costs. The main result in this paper is obtained on sequences of queue operations in which MIN returns nondecreasing values. Such operation sequences characterize the applications mentioned above. We improve the asymptotic running times obtained in [10,11] in cases where intialization dominates because the priority domain is large compared to the length of the operation sequence. Our queues (which are unbalanced) are simpler and use less space, a significant consideration when the domain of priorities is large compared with the length of the sequence of queue operations.

---

†Unless specified otherwise, log is $\log_2$ and ln is the natural logarithm.

In the following sections we consider priority queues which support the operations defined below, where Q is a priority queue, x is the name of a queue element, and p is a priority drawn from a set P with a total order.

```
       INIT(Q) ::=  $Q \leftarrow \emptyset$

  INSERT(p,Q) ::=  let $x \notin Q$
                    $\pi(x) \leftarrow p$
                    $Q \leftarrow Q \cup \{x\}$
                    return(x)

  DELETE(x,Q) ::=  $Q \leftarrow Q - \{x\}$

        MIN(Q) ::=  let x satisfy $\pi(x) = \min_{y \in Q} (\pi(y))$
                    DELETE(x,Q)
                    return(x,$\pi(x)$)

UPDATE(x,p,Q) ::=  if $p < \pi(x)$ then $\pi(x) \leftarrow p$ .
```

Let $\sigma(Q)$ be a sequence of the above operations, on a given priority queue Q, which contains exactly one INIT(Q) operation (at the beginning). The sequence $\sigma(Q)$ will contain n INSERT operations, u UPDATE operations, and m DELETE and MIN operations taken together. Thus the length of $\sigma(Q)$, denoted $|\sigma(Q)|$, is $1 + n + u + m$. We are concerned with the cost of $\sigma(Q)$ on a uniform cost RAM as a function of the variables n, u, and m. Our discussion will be confined to sequences for which $m \leq n$. For the purposes of asymptotic analysis of the cost of $\sigma(Q)$, families of sequences must be considered in which fixed functional relationships between m, n, and u are preserved as, say, n grows. When considered in the context of a family of sequences for which $n + u = O(m)$, a sequence will be called *balanced*. Otherwise, if $n + u = O(m)$ does not hold for the family of sequences, a member sequence will be called *unbalanced*.

UNBALANCED QUEUES FOR ARBITRARY PRIORITY DOMAINS

In all of the balanced queues cited in the preceding section, the data structure used is some variant on a binary tree. Perhaps ternary nodes are permitted in the tree and, perhaps, queue elements appear only at leaf nodes instead of at every node in the tree. The 2-3 trees of [1], for example, employ both of these variations. Trees with nodes of fixed, small degree are asymptotically optimal for balanced operation sequences. The β-heap of [7] is a full tree of degree $\beta \geq 2$, that is, a heap-ordered tree in which every node except possibly one is of degree 0 or β and all nodes of degree less than β are on the two lowest levels of the tree. In a β-heap, restoring heap order in DELETE and MIN requires movement of an element toward the leaves of the tree, so these operations may require as many as $(\beta - 1) \lfloor \log_\beta n \rfloor$ comparisons. The operations INSERT and UPDATE move elements rootward, requiring at most $\lfloor \log_\beta n \rfloor$ comparisons. Initialtion of a β-heap is of constant cost and requires no comparisons. Let the number of comparisons executed on $\sigma(Q)$ be $C(\sigma(Q))$. We conclude from the above discussion that, for a β-heap,

$$C(\sigma(Q)) \leq ((\beta - 1)m + n + u)\log_\beta n .$$

This bound is attainable, to within a uniform constant factor, for all allowable values of m, n, and u. If we let $\beta = \lfloor n^{1/r} \rfloor$ for $r \geq 1$ then the bound becomes

$$C(\sigma(Q)) \leq r(m(n^{1/r} - 1) + n + u) . \qquad (1)$$

It should be noticed that the bound in (1) reduces to $(m + n + u)\log n$ for $r = \log n$, that is, for the balanced queue with $\beta = 2$. For sufficiently unbalanced sequences $\sigma(Q)$ other choices for r are better. For instance, for fixed $\epsilon > 0$, if $m = n > 2^{1.44/\epsilon}$ and $u = n^{1+\epsilon}$ then as n grows the r which optimizes (1) approaches

$$r = \frac{1}{\epsilon - \frac{\log\ln n}{\log n}} .$$

For $n \leq 2^{1.44/\epsilon}$ choosing $r = \log n$ is near to the optimal choice. Priority queues with $m = n$ arise in computing minimum spanning trees and shortest paths in graphs. If the graphs in question have at least $n^{1+\epsilon}$ edges for some fixed ϵ, where n is the number of vertices, the above discussion may be applied (see [6] and [7]). It may be seen that unbalanced queues will be of value in practical cases for values of ϵ no smaller than, say, 1/10. When $\epsilon = 1/5$, for $n = 2^{16}$ the optimal choice of r in (1) gives about one half the number of comparisons obtained when $r = \log n = 16$. If $\epsilon = 1/8$, the improvement is roughly a factor of 2/3 for $n = 2^{16}$.

When $r = \log\log n$, the bound in (1) becomes

$$C(\sigma(Q)) \leq (\log\log n)(m(n^{1/\log\log n} - 1) + n + u) .$$

We notice that the cost of INSERT and UPDATE is of the order of the bound (measured in terms of domain size s) achieved in the best known results for queues over limited priority domains. This improvement, however, is obtained at a cost for MIN and DELETE which grows considerably faster than $\log n$.

We close this section by considering a recursive construction in which the β offspring of each non-leaf node in a β-heap with $\beta = \lfloor n^{1/k} \rfloor$ for fixed k are themselves organized as a γ-heap with $\gamma = \lfloor \beta^{1/k} \rfloor$ whenever $\beta > 2$. The number of comparisons $C(n)$ required for a single DELETE or MIN satisfies the recurrence

$$C(n) \leq \begin{cases} 1 & n \leq 2 \\ kC(n^{1/k}) & n > 2 , \end{cases}$$

the solution to which is $C(n) \leq \log n$. However, INSERT and UPDATE become more costly. In fact, the number of comparisons for each of these operations may be seen to be described by the same recurrence, so there appears to be no advantage to this recursive structure. The trees induced by this recursive construction, however, are not binary. They contain a few nodes of large degree.

QUEUES ON THE PRIORITY DOMAIN {1,...,s}

When s, the size of the restricted priority domain P, is large compared to $n + m + u$, the results of the previous section will be superior to the methods in this section or in [10,11]. However, when s is relatively small it can be advantageous to use a priority queue in which the cost of individual queue operations grows with s rather than n. The algorithm of this section is restricted to nondecreasing sequences $\sigma(Q)$, those for which if some instance of MIN returns the priority q then $q \le p$ for any operation INSERT(p) which follows this instance of MIN in σ. (We will throughout the remaining discussion omit the argument Q when a certain queue is understood.)

Given the domain $P = \{1,\ldots,s\}$, our unbalanced queue employs a sequence of at most $\lfloor \log s \rfloor + 2$ buckets with indices i, $0 \le \text{left} \le i \le \text{right} = \lfloor \log s \rfloor + 1$, where left and right are variables which define the range of active buckets. Bucket i has a size equal to 2^j for some j satisfying $0 \le j < i$ and for all i, left $< i <$ right, bucket i is smaller than bucket $i + 1$. Taken together, the buckets cover the set $\{q,\ldots,s\}$ where q is either the value of the last priority returned by a MIN operation or one if no MIN has yet been executed.† The operations INSERT and UPDATE are performed by a binary search over the buckets and so cost $O(\log\log s)$. MIN requires a search of buckets left, left + 1,... until a nonempty bucket i is discovered. Bucket i, of size 2^j, is replaced in the execution of MIN by a subset of the sequence of buckets of sizes $1,1,2,4,\ldots,2^{j-1}$ chosen so that the smallest priority in bucket i will fall into one of the new buckets of size one. The elements from bucket i are placed, using binary search, into the appropriate new bucket. MIN then completes by deleting and returning an element of smallest priority from the leftmost bucket of size one.

We will analyze the complexity over the sequence σ. Each MIN will require $O(\log s)$ time to find the nonempty bucket i plus $O(\log s)$ to construct the bucket refinement, knowing the minimum element in the original bucket i. Finding the minimum element plus restributing the elements from bucket i to the new buckets will cost $O(n \log s \log\log s)$ over σ since each of the n elements ever to enter the queue is always moved to a bucket no larger than half the size of the bucket it occupied before. For buckets of size one, finding the minimum element is of constant cost. Of course, a bucket of size one will not be refined.

The procedures described are implemented below, in order to verify the following theorem.

THEOREM. A priority queue can be constructed over which the cost of all nondecreasing sequences is $O(n \log s \log\log s + u \log\log s)$ where a fixed functional relationship is assumed between n and u.

Proof: The complexity argument is outlined in the preceding discussion. We note that no multiplications or divisions are required by our algorithms if a table of the values 2^i for $0 \le i \le \lfloor \log s + 1 \rfloor$ are precomputed at a cost of $O(\log s)$. Thus our analysis applies on a RAM under the uniform cost criterion. We omit correctness arguments. These can be constructed easily from the discussion given above. ☐

COROLLARY. Minimum spanning trees and all shortest paths from a single source in undirected graphs with n vertices, e edges, and edge weights drawn from $\{1,\ldots,s\}$ can be found in $O(n \log s \log\log s + e \log\log s)$ time.

---

†$\{q,\ldots,q + s - 1\}$ can easily be covered at no increase in complexity.

Proof: The result follows directly from the Theorem. See [6,7] for details. □

An interesting method for finding shortest paths with running time of $O(nks^{1/k} + e(\log k + 1))$ has been given in [4]. As may be seen, this bound is superior for fixed k when s is small and e is large. These are the conditions, however, when the results of the previous section might be superior to either method. The bound in [4] reduces to the bound for the algorithm described in [5] and [12] when $k = 1$ and to our bound in the Corollary when k is replaced by $\log s$.

In the procedures below, a facility is assumed which supplies new queue nodes x with link fields with which to chain nodes together in buckets, a field bucket(x) which contains the index of the bucket to which x belongs, and $\pi(x)$, the priority of x. The array ℓ contains bucket boundaries; $\ell(i)$ is the least integer which can be contained in bucket i. The array c is maintained so that $c(i)$ is the number of nodes in bucket i.

```
procedure INIT
   left ← min $\{i + 1 | 2^i \geq s\}$
   right ← left
   $\ell$(left) ← 1
   $\ell$(left + 1) ← $2^{left-1} + 1$
   for i ← 0 to left + 1 do  c(i) ← 0  end
end INIT

procedure PLACE(x,i,j)
   let k , $i \leq k \leq j$ , satisfy  $\ell(k) \leq \pi(x) < \ell(k + 1)$
   put node x in bucket k
   bucket(x) ← k
   c(k) ← c(k) + 1
end PLACE

procedure INSERT(p)
   let x be an unused node
   $\pi(x)$ ← p
   PLACE(x,left,right)
   return(x)
end INSERT

procedure DELETE(x)
   remove node x from bucket bucket(x)
   c(bucket(x)) ← c(bucket(x)) - 1
   bucket(x) ← -1
end DELETE

procedure UPDATE(x,p)
   if left ≤ bucket(x) ≤ right  and  p < $\pi(x)$  then
      [DELETE(x)
      $\pi(x)$ ← p
      PLACE(x,left,right)]
end UPDATE
```

```
procedure MIN
   while  c(left) = 0  do
      if  left > right  then  [left ← left - 1
                               return(Λ,0)]
      left ← left + 1
   end
   if  ℓ(left + 1) - ℓ(left) > 1  then
      [let  x  minimize  π(x)  in bucket left
      span ← π(left + 1) - π(x)
      let  t  satisfy  ℓ(left + 1) - ℓ(left) = $2^{t+1}$
      i ← left
      while  span > 1  do
         if  span - $2^t$ > 0  then
            [span ← span - $2^t$
            ℓ(left) ← ℓ(left + 1) - $2^t$
            left ← left - 1]
         t ← t - 1
      end
      c(i) ← 0
      ℓ(left) ← ℓ(left + 1) - 1
      move nodes in bucket  i  to a dummy bucket
      for  each  x  in the dummy bucket  do
         PLACE(x,left,i)
      end]
   let  x ∈ bucket left
   DELETE(x)
   return(x,π(x))
end MIN
```

CONCLUSION

It has been shown that unbalanced queues can be more efficient than balanced queues over unbalanced operation sequences, those in which the number of MIN operations is dominated by the number of INSERT and UPDATE operations. In an unbalanced queue over an arbitrary priority domain, the cost of INSERT and UPDATE grows more slowly with n than the cost of MIN. When operation sequences are known to be nondecreasing in the values returned by MIN then there are unbalanced queues in which operation costs grow with s, the size of the domain of priorities (assumed to be a set of consecutive integers). When n is large in relation to s, these queues are superior.

REFERENCES

1. A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, 1974.

2. M. R. Brown, The analysis of a practical and nearly optimal priority queue, *Tech. Rep.* STAN-CS-77-600, Computer Science Department, Stanford University, 1977.

3. C. A. Crane, Linear lists and priority queues as balanced binary trees, Ph.D. Thesis, Computer Science Department, Stanford University (STAN-CS-72-259), February 1972.

4. E. V. Denardo and B. L. Fox, Shortest route methods: 1. Reaching, pruning, and buckets. (Preliminary version), (unpublished), May 1977.

5. R. B. Dial, Algorithm 360: Shortest-path forest with topological ordering, *Comm. ACM* *12*, 11(Nov. 1969), 632-633.

6. D. B. Johnson, Efficient algorithms for shortest paths in sparse networks, J. ACM 24, 1(Jan. 1977), 1-13.

7. D. B. Johnson, Priority queues with update and finding minimum spanning trees, Information Processing Letters 4, 3(Dec. 1975), 53-57.

8. A. Jonassen and O-J. Dahl, Analysis of an algorithm for priority queue administration, BIT 15 (1975), 409-422.

9. D. E. Knuth, The Art of Computer Programming, Volume 3/Sorting and Searching, Addison-Wesley, 1973.

10. P. van Emde Boas, R. Kaas, E. Zijlstra, Design and implementation of an efficient priority queue, Mathematical Systems Theory 10 (1977), 99-127.

11. P. van Emde Boas, Preserving order in a forest in less than logarithmic time and linear space, Information Processing Letters 6, 3(June 1977), 80-82.

12. R. A. Wagner, A shortest path algorithm for edge-sparse graphs, J. ACM 23, 1(Jan. 1976), 50-57.

POLYNOMIAL b-BALANCED TREES[*]

F. LUCCIO and L. PAGLI
University of Pisa, Pisa, Italy

ABSTRACT

The new class of Pbk trees is presented, where height balance is maintained for the nodes lying on particular paths. In such trees the number of nodes is a polynomial function of the height, in the worst case. The class of IPbk trees is then introduced, composed of Pbk trees that can be built by successive node insertions. The average behaviour of IPbk trees is close to the one of height balanced trees: the main advantage of the former trees is a reduced number of restructurings required after insertion, at the cost of a slight increase of search length.

INTRODUCTION

In a previous paper [1] Polynomial k trees (briefly, Pk trees) were introduced, defined as binary search trees where height balance is maintained only for the nodes on the longest paths. In Pk trees, the maximum allowed absolute value of balance for the above nodes is one. As in the theory of height balanced trees [2,3,4], we now extend the definition of Pk trees, by assigning to the bound on node balance a generic integer value $b \geq 1$.

The following terminology will be used. A path is the string of tree nodes from the root to a leaf. An i-path is a path of length i. The balance $B(N)$ of a node N is the difference between the heights of the right and the left subtree of N. h always denote the height of the given tree.

We now define our new trees.

Definition 1. For a pair of integer values $b \geq 1$ and $k \geq 0$, a Polynomial b-balanced k tree (Pbk tree) is a binary tree such that:

for k=0: there exists at least one h-path σ, such that, for any node $N \in \sigma$, $|B(N)| \leq b$;

for k>0: each i-path σ, $h-k+1 \leq i \leq h$, is such that, for any node $N \in \sigma$, $|B(N)| \leq b$.

For any pair of values $b \geq 1$, $k \geq 0$, definition 1 introduces a family of Pbk trees, denoted by $\mathcal{F}(\text{Pbk})$. For b=1, definition 1 coincides with the one of Pk trees given in [1], that is: $\mathcal{F}(\text{P1k}) = \mathcal{F}(\text{Pk})$. (In fact, several results found in [1] for Pk trees can be seen as special cases of the ones given in this paper, for b=1.) For $k \geq h$, any Pbk tree is a height balanced tree of maximum balance b [2].

In the next sections some properties of Pbk trees will be presented, and their behavior will be studied under node insertion.

PROPERTIES OF MINTREES OF $\mathcal{F}(\text{Pbk})$

In this section we study the properties of Pbk trees of minimum number of nodes, for any given height h. As for height balanced trees, this allows to determine the upper bound to the search length for a node, as a function of the number of nodes of the tree.

---

[*] This work was supported in part by Comitato Nazionale per le Scienze Matematiche, C.N.R., under research contract No. 75.01035.01.

Definition 2. The mintree min (Pbk,h) of $\mathcal{F}$(Pbk) is a Pbk tree of height h and the minimum number of nodes, such that $B(N) \leqslant 0$ for any node N.

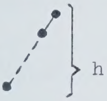
Note that, among all possible trees of height h with minimum number of nodes, we give the name of mintrees to the ones with balance $B(N) \leqslant 0$ for every node N. In fact, for any given h, min(Pbk,h) is unique, and has the form given in the following theorem.

Theorem 1.

min(Pbk,0) = empty;

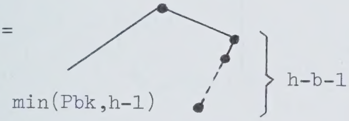
(i)

min(Pbk,h)=



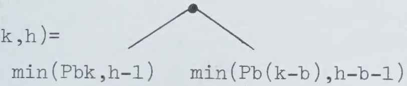
for $1 \leqslant h \leqslant b$;

(ii) min(Pbk,h)=



for $0 \leqslant k \leqslant b$ and $h > b$;

(iii) min(Pbk,h)=



for $k > b$ and $h > b$.

Proof. Statements (i) are obvious. To prove the subsequent statements, letting λ and ρ respectively indicate the left and right subtree of the root, we note that any mintree obviously has exactly one h-path lying in λ. To prove statement (ii), note that λ must be a mintree of height h-1 and same value of k. ρ must have height h-b-1, $b \geqslant k$, hence ρ is not traversed by any i-path, $h-k+1 \leqslant i \leqslant h$. Therefore, ρ is the tree of height h-b-1 with minimum number of nodes, that is a simple chain of h-b-1 nodes. To prove statement (iii), note that λ must be a mintree of height h-1 and same value of k. Obviously, B(root)=-b, that is ρ has height h'=h-b-1; and all the i-paths traversing ρ, $h-k+1 \leqslant i \leqslant h-b$, must have all the nodes N with $B(N) \leqslant b$. The portions of such i-paths local to ρ are j-paths, j=i-1, $h-k \leqslant j \leqslant h-b-1$. Therefore we have: $h'-(k-b)+1 \leqslant j \leqslant h'$, that is ρ=min(Pb(k-b), h-b-1). ▼

Examples of mintrees are given in fig. 1.

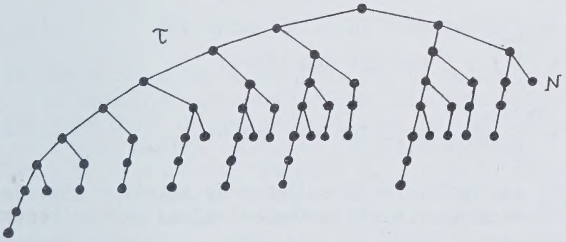


Fig. 1 - Examples of Pbk mintrees.
τ=min(P2 5,10)=min(P2 6,10).
τ-{N}=min(P2 3,10)=min(P2 4,10).

Some corollaries follow from theorem 1.

Corollary 1.1. $\min(Pbk_1,h)=\min(Pbk_2,h)$ for any pair k_1,k_2 such that either $0\leq k_1\leq k_2\leq b$, or $pb+1\leq k_1\leq k_2\leq (p+1)b$ for any integer $p\geq 1$.

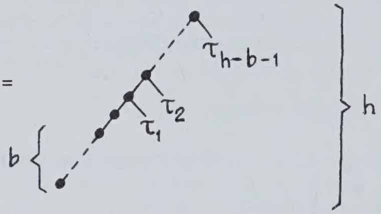
Proof. For $0\leq k_1\leq k_2\leq b$, the statement immediately follows from points (i) and (ii) of theorem 1. For $pb+1\leq k_1\leq k_2\leq (p+1)b$, the statement follows from point (iii) of the theorem, and the result just found for $0\leq k_1\leq k_2\leq b$. ▼

Corollary 1.1 can be checked on fig. 1, where $\tau=\min(P2\ 5,10)=\min(P2\ 6,10)$ $(b=2,\ p=2,\ k_1=5$ and $k_2=6)$; and $\tau-\{N\}=\min(P2\ 3,10)=\min(P2\ 4,10)$ $(b=2,\ p=1,\ k_1=3$ and $k_2=4)$.

The following corollary immediately derives from theorem 1.

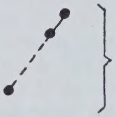
Corollary 1.2. For $h\geq b+2$:

$\min(Pbk,h)=$



where, $1\leq i\leq h-b-1$:

$\tau_i=$



, for $k\leq b$;

$\min(Pb(k-b),i)$, for $k>b$.

Let $n_{b,k,h}$ denote the number of nodes of $\min(Pbk,h)$. From corollary 1.1 it immediately follows:

$$n_{bk_1h}=n_{bk_2h}\ ,\quad \begin{array}{l} 0\leq k_1\leq k_2\leq b \quad \text{or} \\ pb+1\leq k_1\leq k_2\leq (p+1)b \text{ for any integer } p\geq 1. \end{array} \qquad (1)$$

From theorem 1 and corollary 1.2 it follows:

$$n_{b,k,h}=\begin{cases} h, & \text{for any } k \text{ and } 0\leq h\leq b+1; \qquad (2) \\ h+\sum\limits_{i=1}^{h-b-1} i\ , & \text{for } k\leq b \text{ and } h>b+1; \qquad (3) \\ h+\sum\limits_{i=1}^{h-b-1} n_{b,k-b,i}, & \text{for } k>b \text{ and } h>b+1. \qquad (4) \end{cases}$$

The values of $n_{b,k,h}$ can be directly computed by relation (3), for $k\leq b$, and can be derived from previously computed values by the recurrence equation (4), for $k>b$.

Deriving a general expression for $n_{b,k,h}$ as a function of b,k and h, is a difficult task. In the following two corollaries we give such an expression for $k\leq b$, and the asymptotic values of $n_{b,k,h}$ for increasing h and for any b,k.

Corollary 1.3. For $h \geq b+2$ and $k \leq b$:

$$n_{b,k,h} = \frac{1}{2}\left(h^2-(2b-1)h+b^2+b\right).$$

Proof. It immediately follows from relation (3), making use of the known formula for $\sum_{i=1}^{n} i$ and elementary algebraic transformations. ▼

Corollary 1.4. For any couple $b,k,k \geq 1$:

$$\left[n_{b,k,h} = \frac{1}{(p+2)!}\, h^{p+2}\right]_{h\to\infty} \quad \text{where } p \text{ is such that } pb+1 \leq k \leq (p+1)b.$$

Proof. By induction on p. The thesis holds for $p=0$, that is for $1 \leq k \leq b$, by corollary 1.3. Assume that the thesis holds for $p=\bar{p}$, that is for any value $k=k(\bar{p})$, $\bar{p}b+1 \leq k(\bar{p}) \leq (\bar{p}+1)b$:

$$\left[n_{b,k(\bar{p}),h} = \frac{1}{(\bar{p}+2)!}\, h^{\bar{p}+2}\right]_{h\to\infty} \quad ;$$

we prove that it holds for $p=\bar{p}+1$, that is for any value $k=k(\bar{p}+1)$, $(\bar{p}+1)b+1 \leq k(\bar{p}+1) \leq (\bar{p}+2)b$.

From equation (4) we have:

$$n_{b,k(\bar{p}+1),h} = h + \sum_{i=1}^{h-b-1} n_{b,k(\bar{p}+1)-b,i} = h + \sum_{i=1}^{h-b-1} n_{b,k(\bar{p}),i},$$

that is, for $h\to\infty$:

$$\left[n_{b,k(\bar{p}+1),h} = h + \frac{1}{(\bar{p}+2)!} \sum_{i=1}^{h-b-1} i^{\bar{p}+2} = \frac{1}{(\bar{p}+2)!}\, \frac{(h-b-1)^{\bar{p}+3}}{\bar{p}+3} = \right.$$

$$\left. = \frac{1}{(\bar{p}+3)!}\, h^{\bar{p}+3}\right]_{h\to\infty}$$

where, at any step in the chain of equalities, the only terms of highest degree are retained. ▼

From corollary 1.4 and Stirling's approximation of the factorial, we derive the asymptotic value of the height of a Pbk tree, $k \geq 1$, as a function of the number of nodes:

$$\left[h = \left((p+2)!\, n_{b,k,h}\right)^{\frac{1}{p+2}} \cong \frac{p+2}{e}\, n_{b,k,h}^{\frac{1}{p+2}}\right]_{n_{b,k,h}\to\infty} \tag{5}$$

for the value p such that:

$pb+1 \leq k \leq (p+1)b$.

As already noted, for $k\to h$ a Pbk tree tends to become a height balanced tree of maximum balance b. In fact, if $k\to h\to\infty$ in relation (5), h becomes a logarithmic function of $n_{b,k,h}$.

Pbk TREES BUILT BY NODE INSERTIONS

We now discuss how a Pbk tree can be built by successive insertions of nodes. We recall a known definition [1], namely: for any path σ, the critical node $CR(\sigma)$ is the node closest to the leaf of σ, among all the nodes of σ with maximal absolute value of balance.

Let N by a new node, inserted in a tree τ, and let σ be the i-path containing N. τ must be rebalanced after such an insertion if and only if $|B(CR(\sigma))|>b$ and $i \geq h-k+1$ (definition 1). Rebalancing is performed on $CR(\sigma)$ by standard rotations, using the following insertion procedure, which is an immediate extension of the one used for AVL trees[5], and for Pk trees[1].

```
procedure INSERTION (N,τ);
    traverse τ until the location for N is reached;
    insert N, and consequently update the balances of the ancestors of N;
```

(let σ be the i-path containing N)
if ($|B(CR(\sigma))|>b$ and $i\geq h-k+1$) then rebalance τ on $CR(\sigma)$.

From the properties of standard rotations it immediately follows that, for any given pair $k=\bar{k}$, $b=\bar{b}$, if INSERTION is used to insert N in $\tau_1 \in \mathfrak{F}(P\bar{b}\bar{k})$, it generates $\tau_2 \in \mathfrak{F}(P\bar{b}\bar{k})$. From the above discussions it follows that the cost of a rebalancing operation is the same for Pbk trees and AVL trees.

For any pair b,k, the new family of trees $\mathfrak{F}(IPbk)$ is now defined.

Definition 3. An Insertion Polynomial b-balanced k tree of n nodes (IPbk tree) is a Pbk tree built by n successive applications of the procedure INSERTION.

That is, an IPbk tree τ is built through a sequence of trees ρ_0=empty, $\rho_1,\ldots,\rho_n=\tau$, such that: $\rho_i \in \mathfrak{F}(Pbk)$, $0\leq i\leq n$.

An immediate consequence of definitions 1 and 3 is that: $\mathfrak{F}(Pbk) \supseteq \mathfrak{F}(IPbk)$. Let us now study the properties of mintrees of $\mathfrak{F}(IPbk)$.

Lemma. $\min(IPbk,h) \in \mathfrak{F}(Pb(k+1))$.

Proof. If a tree of height h has been built by successive applications of INSERTION, it must have firstly reached the height h by insertion of a node at level h in a tree of height h-1. Consequently, $\min(IPbk,h)$ must have exactly one node (say N) at level h, which has been inserted last, in $\min(IPbk,h)-\{N\}$. Since $\min(IPbk,h)-\{N\} \in \mathfrak{F}(Pbk)$ by definition 3, and such a tree has height h-1, then $\min(IPbk,h) \in \mathfrak{F}(Pb(k+1))$ by definition 1. ▼

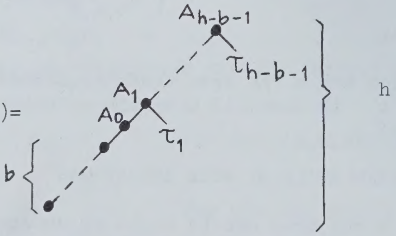
Theorem 2.

$$\min(IPbk,h) = \min(Pb(k+1),h).$$

Proof. We divide the proof in two parts, namely, we prove that: (i) $\min(Pb(k+1),h) \in \mathfrak{F}(IPbk)$, that is, such a tree can be built as a Pbk tree; (ii) $\min(Pb(k+1),h)$ is the mintree of height h of $\mathfrak{F}(IPbk)$, that is, $\min(Pb(k+1),h)=\min(IPbk,h)$.

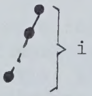
(i) By induction on h, we prove that $\min(Pb(k+1),h+1)$ can be built as a Pbk tree by inserting nodes in $\min(Pb(k+1),h)$. That is, $\min(Pb(k+1),h+1)$ can be built through a sequence $\rho_0=\min(Pb(k+1),h),\rho_1,\ldots,\rho_n=\min(Pb(k+1),h+1)$, $\rho_i \in \mathfrak{F}(Pbk)$ and ρ_i is of height h, $0\leq i\leq n-1$. For $h\leq b+1$ the induction is trivially satisfied (see theorem 1, point (i)). For $h>b+1$, by corollary 1.2 we have:

$\min(Pb(k+1),h)=$



where, $1\leq i\leq h-b-1$:

$\tau_i =$



, for $k+1\leq b$;

$\min(Pb(k+1-b),i)$, for $k+1>b$.

For $k+1 \leqslant b$, min(Pb(k+1),h+1) can be built from min(Pb(k+1),h) by inserting a new node as right son of A_0, and by adding to each τ_i a new node at level h-b+1, thus generating a sequence ρ_0=min(Pb(k+1),h), $\rho_1, \dots, \rho_{n-1}$ of Pbk trees of height h (in fact, in such trees all the nodes in the last $k \leqslant b-1$ levels lie on the (unique) h-path, all of whose nodes are balanced); finally the node at level h+1 is inserted.

For $k+1>b$, min(Pb(k+1),h+1) can be built from min(Pb(k+1),h) by inserting a new node as right son of A_0, and by adding new nodes to each τ_i= min(Pb(k+1-b),i), to transform it into τ_{i+1}= min(Pb(k+1-b),i+1). By the inductive hypothesis, such a transformation can be realized through the sequence $\tau_i=\lambda_{i_0}, \lambda_{i_1}, \dots, \lambda_{i_{m-1}}, \lambda_{i_m}=\tau_{i+1}$, with $\lambda_{i_j} \in \mathcal{F}$(Pb(k-1)) and λ_{i_j} of height i, $0 \leqslant j \leqslant m-1$. This generates the global sequence ρ_0=min(Pb(k+1),h), $\rho_1, \dots, \rho_{n-1}$, where $\rho_i \in \mathcal{F}$(Pbk), $0 \leqslant i \leqslant n-1$, since the nodes in the last k levels of such trees either lie in the (unique) h-path, all of whose nodes are balanced; or lie in the last k-b levels of a Pb(k-b) tree λ_{i_j};or lie in the last k-b+1 levels of a Pb(k-b+1) tree τ_{i+1}. Finally, the node at level h+1 is inserted, to form min(Pb(k+1),h+1). We conclude that min(Pb(k+1),h) $\in$ $\mathcal{F}$(IPbk).

(ii) By the lemma, min(IPbk,h) must be found in $\mathcal{F}$(Pb(k+1)); since min(Pb(k+1),h) is buildable as a Pbk tree (part (i)), then min(IPbk,h)= min(Pb(k+1),h). ▼

As an immediate consequence of theorem 2 and corollary 1.1, we have:
Corollary 2.1. min(IPbk_1,h)=min(IPbk_2,h), for any pair k_1, k_2 such that $qb \leqslant k_1 \leqslant k_2 \leqslant qb+b-1$, for any integer $q \geqslant 0$.

For example, in fig. 1 we have:
τ=min(P2 5,10)=min(IP2 4,10)=min(P2 6,10)=min(IP2 5,10)
(theorem 2, and corollary 2.1 with q=2); and
τ-{N}=min(P2 3,10)=min(IP2 2,10)=min(P2 4,10)=min(IP2 3,10)
(theorem 2, and corollary 2.1 with q=1). Recalling the proof of the lemma, we note that the node at level 10 must have been the last inserted node.

Letting $n_{I,b,k,h}$ denote the number of nodes of min(IPbk,h), from theorem 2 immediately follows $n_{I,b,k,h}=n_{b,k+1,h}$, and relations (2), (3) and (4) can be applied to determine such values. In particular, relation (5) becomes:

$$\left[h \cong \frac{q+2}{e} \, n_{I,b,k,h}^{\frac{1}{q+2}} \right]_{n_{I,b,k,h} \to \infty} \tag{6}$$

for the value q such that: $qb \leqslant k \leqslant qb+b-1$.

EXPERIMENTAL RESULTS ON IPbk TREE PERFORMANCE

The asymptotic value of the worst case search length for a node in a IPbk tree is given in relation (6). Such a value decreases for increasing k, in fact, for $k \to h$, IPbk trees tend to become height balanced trees. (Note that, for $k \to h \to \infty$ in relation (6), h becomes a logarithmic function of $n_{I,b,k,h}$).

The above asymptotic value of h applies to the mintrees, however, in practical applications the average value of h for the trees actually built may be a more important parameter. Furthermore, an even more important parameter is the average search length for all the nodes.

In order to evaluate the above parameters, an extensive set of simulation runs has been performed. Groups of 500 IPbk trees of 1000, 2000, 3000, 4000 and 5000 nodes have been randomly generated by node insertion, for b=1,2,3,4,5 and k=0,1,2,3,4. In addition, groups of 500 height balanced trees of 1000, 2000, 3000, 4000 and 5000 nodes have been randomly generated by node insertion, for b=1,2,3,4,5, and the results have been compared with the ones for IPbk trees. (The results for height balanced trees are in exact agreement with the ones given in [4]). For brevity, only the results for trees of 5000 nodes will now be presented.

In table I we give the average values of h, for IPbk trees and height balanced trees.

b \ k	0	1	2	3	4	height balanced
1	16.11	14.99	14.85	14.78	14.78	14.78
2	17.40	16.69	16.26	16.14	16.14	16.14
3	18.56	18.49	18.18	17.92	17.86	17.86
4	19.68	19.58	19.52	19.42	19.34	19.34
5	20.71	20.63	20.62	20.60	20.46	20.38

Table I. Average value of maximal search length for IPbk trees and height balanced trees of 5000 nodes.

In table II we give the average value of search length for all the nodes, computed as the average number of comparisons to find all the nodes already present in the tree.

b \ k	0	1	2	3	4	height balanced
1	12.59	11.92	11.67	11.61	11.59	11.59
2	12.97	12.65	12.21	12.09	12.06	12.06
3	13.25	13.01	12.73	12.55	12.47	12.44
4	13.51	13.31	13.09	12.94	12.85	12.78
5	13.74	13.56	13.39	13.27	13.18	13.11

Table II. Average value of search length for IPbk trees and height balanced trees of 5000 nodes.

To evaluate the performance of the insertion process a final parameter has been computed, namely, the average number of rotations. The values of this parameter are given in table III.

From the examination of all the above results we conclude that the effect of using increasing values of b is similar in IPbk trees and height balanced trees, namely, the number of rotations is decreased at the cost of an increasing value of search length. The effect of increasing k is instead the one of a relevant increase of the number of rotations, however, the search length decreases on the average. For k=4, IPbk trees of up to 5000 nodes are almost coincident with height balanced trees.

b \ k	0	1	2	3	4	height balanced
1	0.091	0.289	0.423	0.461	0.466	0.465
2	0.047	0.091	0.162	0.196	0.206	0.207
3	0.027	0.046	0.072	0.098	0.119	0.120
4	0.017	0.026	0.040	0.054	0.065	0.074
5	0.011	0.016	0.024	0.032	0.039	0.047

Table III. Average number of rotations for a node inserted in IPbk trees and height balanced trees, computed on trees of 5000 nodes.

CONCLUSION

For any given pair of integers b,k the new family of IPbk trees has been introduced, where, in the worst case, the height is of the order of the (q+2)th root of the number of nodes, and q is approximately proportional to k/b. For increasing values of k, such trees tend to become height balanced trees with maximum allowed unbalancing of b.

IPbk trees are built by node insertion, and consequent rebalancing (if needed) is performed by an immediate extension of the standard procedure for height balanced trees. Tree rebalancing is therefore performed in constant time.

The main advantage of IPbk trees over height balanced trees is the reduced average number of restructurings after insertion, at the cost of a slight increase of the average search length. The problem of node deletion is rather complex and has been left open, however, in most applications deletion is not required, or can be performed once in a while by restructuring the whole tree.

REFERENCES

[1] F. Luccio and L. Pagli. Polynomial trees. Istituto di Scienze dell' Informazione, University of Pisa, Italy. Technical Report S-77-12, April 1977.(Submitted for publication).

[2] C.C. Foster. A generalization of AVL trees. Comm. ACM 16, 8 (Aug. 1973), 513-517.

[3] F. Luccio and L. Pagli. On the height of height balanced trees. IEEE Transactions on Computers C-25, 1 (Jan. 1976), 87-90.

[4] P.L. Karlton, S.H. Fuller, R.E. Scroggs and B. Kaeler. Performance of height balanced trees. Comm. ACM 19, 1 (Jan. 1976), 23-28.

[5] D.E. Knuth. The Art of Computer Programming, Vol 3. Addison-Wesley, Reading, Mass., 1973.

DATA ARRANGEMENT IN MAGNETIC BUBBLE MEMORIES

C.K. WONG
IBM T. J. Watson Research Center
P.O. Box 218, Yorktown Heights, N.Y. 10598

Abstract: In this paper we discuss two kinds of magnetic bubble memories. Because of their serial nature, average access time tends to be long. To improve upon this, some additional features are introduced to these memories. The problem of arranging data to a given order (i.e. to generate arbitrary permutation) is also discussed. Some proposed algorithms are shown to be optimal while others are just near-optimal. Some open problems are also mentioned at the end.

I. TWO KINDS OF MAGNETIC BUBBLE MEMORIES

The first kind of bubble memory discussed here is the so-called T-I bar memories and their later variants. Bit information is represented by the presence or absence of a bubble. Conceptually, it can be regarded as a long loop of cells as far as data accessing is concerned. Data items are stored in the cells. The only way to access an item is through an I/O port, i.e., the item has to be in a specific location (location 0 in Fig. 1) before it can be accessed [2].

The second kind is the so-called bubble lattice files [3,4]. Bit information is encoded in the wall states of the bubbles, resulting in higher density. Conceptually, it can be regarded as a rectangular array of bubbles with buffer zones on both ends. (Fig. 2) Bubbles can be access through a combination of two operations: (1) lattice translation and (2) column shifting.

II. DATA ARRANGEMENT IN FIRST KIND OF BUBBLE MEMORY (T-I BAR)

In this kind of memory, the data items can be circulated along the loop, say, in a counterclockwise direction. Then if the size of the loop is n, the average access time will be n/2.

To improve upon this, we notice that in general access frequencies of the items are not the same. To take advantage of this, one would like to keep the more frequently accessed items closer to the I/O port. Furthermore, one would like to achieve this automatically without prior knowledge of the access frequencies. One way to approximate this is to keep the latest accessed item at the I/O port and shift the other items accordingly. For example, suppose item i is located at location i for all i, and we want to access item k. Afterwards we would like to maintain the order of items in the following way: $k,0,1,\ldots,k-1, k+1,\ldots,n-1$ in a clockwise direction. This will be referred to as the Last Use Ordering.

In the long run, more frequently accessed items will tend to be closer to location 0. To accomplish such an ordering, we introduce a new operation to the memory. We therefore have the following two kinds of operations: (a) and (e). (Fig. 3)

To access an item at location k, the optimal sequence is as follows, where optimality is measured in terms of the total number of operations.

(1) For $1 \le k \le \left[\frac{6n-7}{8}\right\rfloor$, apply $a^k e^k$. (Total = $2k$).

(2) For $\left\lfloor\frac{6n-7}{8}\right\rfloor \le k \le n-2$, apply $(eae)^{n-k-1} ea (aea)^{n-k-2}$.
(Total = $6(n-k)-7$).

(3) For $k=n-1$, apply eae. (Total = 3).

The next question is how to realize a general permutation of items in memory. No optimal algorithm is known. But the following algorithm differs from optimum by only a constant factor. To illustrate its operation, we use a simple example. Suppose the required permutation is given in Fig. 4.

We shall use the previous accessing algorithm to bring item 3 to location 0 first. Then apply a, so that 3 is at location 6. (See Figure 5). Then repeat the same process by bringing item 2 to location 0, and so on.

It can be shown [2] that for both the worst-case and the average case $O(n^2)$ operations are needed to realize an arbitrary permutation. In fact, $O(n^2)$ can also be shown to be a lower bound for both cases. Thus, this algorithm differs from optimum by only a constant factor.

More recently, another set of simple operations has been implemented. They are diagramatically shown in Fig. 6. Consequently, the number of operations to access an item at location k and to maintain the Last Use Ordering can be further cut down.

In general, the optimal sequence of operations to access an item at k can be described as follows ($n=2m+1$, $m \geq 1$):

(1) For $1 \leq k \leq m$, apply $a^{k-1}\, d\, b^{k-1}\, c$. (Total = $2k$).

(2) For $m+1 \leq k \leq n-2$, apply $c\, a^{n-k-1}\, d\, b^{n-k-2}$. (Total$=2(n-k)-1$).

(3) For $k = n-1$, apply cac. (Total = 3).

To generate arbitrary permutations, one can devise an algorithm utilizing this optimal sequence as done before. It takes $O(n^2)$ operations for both the worst and average cases. The previous lower bounds hold here also.

III. DATA ARRANGEMENT IN SECOND KIND OF BUBBLE MEMORY (BLF)

In this memory, we use a column as a basic unit of information. Instead of Last Use Ordering, we shall implement another kind of ordering, called Transposition Ordering which also brings more frequently accessed items closer to the access channel[3]. It can be described as follows:

Suppose we have a list of n items. The one located in the access position is called $C(1)$, the next $C(2)$, then $C(3)$, etc. Transposition Ordering means that every time an access is made to $C(k)$, it changes position with $C(k-1)$ unless $k = 1$.

To implement this ordering in BLF, we need the following modifications. First of all, the access channel consists of a shift register that can propagate two adjacent bubble columns at the same time. (See Fig. 7.) Propagation can be accomplished with the same technology as in a conventional BLF by isolating two columns instead of one from the rest of the lattice. Secondly, the access channel is terminated at one end by a pair of write stations (W,W') and at the other by a pair of read stations (R,R'). The write atations are connected to a switch that has two modes: SET and RESET. In RESET mode, the bubble sensed at R is regenerated at W, thus operating in a manner similar to the conventional BLF (and likewise for R' and W'). In SET mode, however, the roles of W and W' are reversed so that the bit that is shifted out through R is shifted back through W' into the adjacent column, and similarly for R' and W. Thus, in the time taken to shift the full length of a column, the two adjacent columns under the access channel are automatically transposed.

To realize an arbitrary permutation of columns in BLF, we extend this idea to the following more general model: We put k + 1 access channels $A_0, A_1, \ldots, A_k$ in a BLF, with the distance between A_0 and A_1 being d_i and $d_1<d_2<\ldots<d_k$, (see Fig. 8, where $d_1 = 1$, $d_2 = 4$, etc.) and assume columns under channels A_0 and A_i, $i\geq 1$, can be interchanged using the switch described above. (Interchanging columns under A_i and A_j, $i,j \neq 0$, is not called for.) Thus, by translating the columns, we can effectively interchange two columns that are distant d_1, $1\leq i\leq k$, from each other. We further assume that the lattice translation time is negligible compared to column interchanges. Thus, the total number of column interchanges is taken as the measure of complexity of the algorithm involved. Our primary objective is to permute the columns of a BLF to a required permutation using the minimum number of interchanges. As in most of such cases, an optimal algorithm is very difficult to find. We

therefore resort to a simple but efficient heuristic.

Simply stated, we number the column locations 1, 2, ...,n from left to right, say. We shall move the column with destination location 1 to its destination first. In general, assume locations 1, 2, ...,k-1 are all filled with correct columns. To move the correct column to location k, we can only do interchanges in the right hand side of the column k-1, i.e. columns 1, 2,..., k-1 are not to be disturbed. It can be shown [4] that this algorithm takes $n\left(\frac{n}{2d_k}+B\right)$ operations for the worst case and $\frac{n}{2}\left(\frac{n}{2d_k}+B\right)$ for the average case, where

$B = \sum_{i=1}^{k-1} \left\lfloor \frac{d_{i+1}}{2d_i} \right\rfloor$. On the other hand, lower bounds for the worst-case and average case can be shown to be $\frac{n}{2}\left(\frac{n}{2d_k}+B\right)$ and $\frac{n}{2}\left(\frac{n}{3d_k}+\frac{B}{2}\right)$ respectively. If we take $d_i = n^{\frac{i-1}{k}}$, then the number of operations for the algorithm is minimized for both cases.

IV. FURTHER REMARKS

Another important feature of the first kind of bubble memory is its stoppability, i.e., we can stop the rotation of the loop and even reverse its direction of rotation at any desirable moment without too much loss in time. We can then consider the following simplified model: Loops are stacked up to form a drum-like storage device, where the tracks are not constrained to rotate in synchronism, i.e., some tracks can stop while others can go forward or backward. For simplicity, just consider the case when stop and forward individual tracks are available. Assume that there are m tracks containing n records each. If we can have one step look-ahead of the request string, i.e., if we know two consecutive requests r_1, r_2 at a time, then we can rotate the track containing r_1 to the I/O port, while shifting the track containing r_2 optimally. The remaining tracks can either rotate with the track containing r_1 or remain intact.

If we assume random access request, these two options are statistically the same and the average access time per record is significantly reduced in comparison to the uncontrolled case [1].

Some open problems remain. For example, what are the optimal algorithms for generating permutations in both kinds of bubble memories? More recently, a 2-dimensional bubble memory has been designed, which can be regarded as a square array of grid points with integer coordinates in a plane. The accessing time can be described by the L_1 metric. What is the optimal algorithm or efficient heuristics to generate permutations in such a situation?

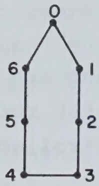


Fig. 1

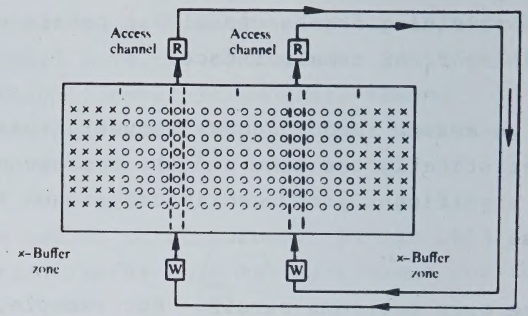
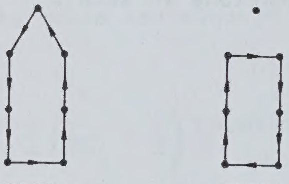

Access channel
R
Access channel
R
x-Buffer zone
W
W
x-Buffer zone

Fig. 2



OPERATION a

OPERATION e

Fig. 3

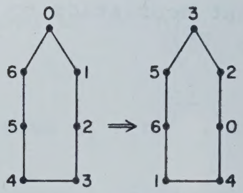

0
6
1
5
2
4
3
3
5
2
6
0
1
4

Fig. 4

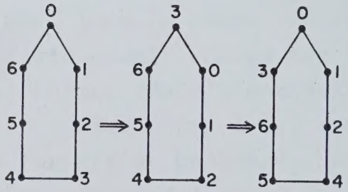
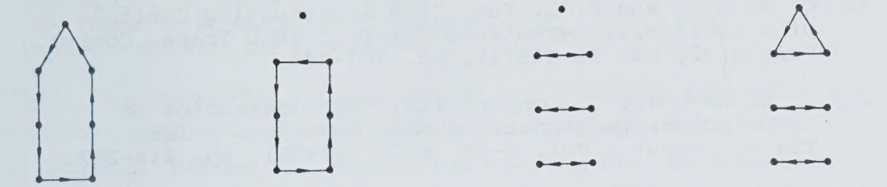

0
6
1
5
2
4
3
3
6
0
5
1
4
2
0
3
1
6
2
5
4

Fig. 5



OPERATION a OPERATION b OPERATION c OPERATION d

Fig. 6

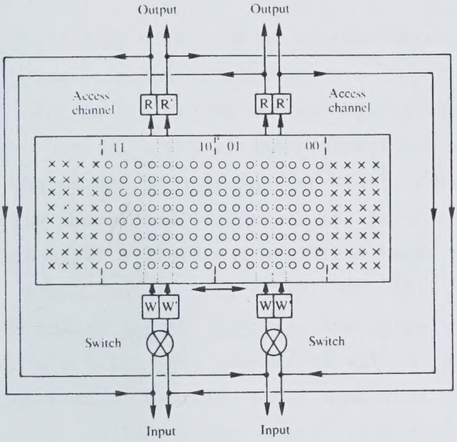

Output
Output
Access channel
R
R'
R
R'
Access channel
11
10
01
00
W
W'
W
W'
Switch
Switch
Input
Input

Fig. 7

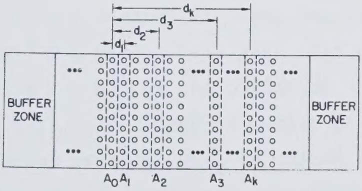

d_k
d_3
d_2
d_1
BUFFER ZONE
BUFFER ZONE
A_0
A_1
A_2
A_3
A_k

Fig. 8

REFERENCES

1. C. K. Wong and P. C. Yue, "The Anticipating Control of a Cyclically Permutable Memory," IEEE Trans. Comput., Vol. C-22, No. 5, (1973), pp. 481-488.

2. C. K. Wong and D. Coppersmith, "The Generation of Permutations in Magnetic Bubble Memories," IEEE Trans. Comput., Vol. C-25, No.3, (1976), pp. 254-262.

3. C. K. Wong and P. C. Yue, "Data Organization in Magnetic Bubble Lattice Files," IBM J. Res. Develop., Vol. 20, No. 6, (1976), pp. 576-581.

4. A. K. Chandra and C. K. Wong, "The Movement and Permutation of Columns in Magnetic Bubble Lattice Files," IBM Res. Report RC 6494, Apr. 1977.

FAST SIMULATION OF COMBINATIONAL LOGIC NETWORKS BY MACHINES WITHOUT RANDOM-ACCESS STORAGE

NICHOLAS PIPPENGER
Mathematical Sciences Department
IBM Thomas J. Watson Research Center
Yorktown Heights, New York 10598

ABSTRACT: A combinational logic network of size N can be represented in several ways by files containing $O(N)$ records of length $O(\log N)$, or a total length of $O(N \log N)$. We show that a machine with one-dimensional tapes can simulate such a network with $O(N \log N)$ operations on records of length $O(\log N)$, or a total time of $O(N(\log N)^2)$. This improves previous bounds of $O((N \log N)^2)$ due to Schnorr and of $O(N(\log N)^3)$ due to Swamy and the author, independently.

NETWORKS

The networks we shall deal with will be acyclic interconnections of inputs, outputs, and gates by means of wires.

Let $\{u(1), \ldots, u(s)\}$ be a finite set of objects called operations, and let each operation $u(r)$ $(1 \leq r \leq s)$ have associated with it a natural number $q(r)$ called the number of operands for $u(r)$. Such a set of operations, together with the associated numbers of operands, will be called a basis. Throughout this paper a fixed but arbitrary basis will be considered.

Let $\{a(1), \ldots, a(n)\}$, $\{b(1), \ldots, b(m)\}$, and $\{h(1), \ldots, h(\ell)\}$ be finite sets of objects called inputs, outputs, and gates, respectively. Let each gate $h(j)$ $(1 \leq j \leq \ell)$ have associated with it an object $c(j)$ called an outlet, an operation $u(r(j))$, and a number of objects $d(j,1), \ldots, d(j,q(r(j)))$ called inlets, where $q(r(j))$ is the number of operands for $u(r(j))$.

Let each output $b(j)$ $(1 \leq j \leq m)$ have associated with it a pair $[b(j), e(j)]$ called a wire, where each $e(j)$ is either an input $(a(i)$ for some $1 \leq i \leq n)$ or an outlet $(c(i)$ for some $1 \leq i \leq \ell)$. Let each inlet $d(j,p)$ $(1 \leq j \leq \ell, 1 \leq p \leq q(r(j)))$ have associated with it a pair $[d(j,p), e(j,p)]$ also called a wire, where each $e(j,p)$ is either an input $(a(i)$ for some $1 \leq i \leq n)$ or the outlet of a "preceding" gate $(c(i)$ for some $1 \leq i \leq j-1)$. The input or outlet of a wire will be called its origin; its output or inlet will be called its termination.

Such sets of inputs, outputs, and gates, together with the associated outlets, operations, inlets, and wires, will be called a network.

The notion of "preceding" that occurred above can be extended in a natural way to all the inputs, outputs, inlets, and outlets of a network, as follows:

$a(1) < \ldots < a(n) < d(1,1) < \ldots < d(1,q(r(1))) < c(1) < \ldots$

$\ldots < d(\ell,1) < \ldots < d(\ell,q(r(\ell))) < c(\ell) < b(1) < \ldots < b(m)$.

Whenever we speak of objects preceding one another, or of their being in sequence, it is this relation to which we shall be referring.

FUNCTIONS

Consider a finite set called the domain. Let each operation $u(r)$, for which there are $q(r)$ operands, have associated with it a domain-valued function of $q(r)$ domain-valued arguments. A domain, together with such an association of functions with operations, will be called an interpretation for the basis. Throughout this paper a fixed but arbitrary interpretation will be considered.

Suppose that a domain value x_j is assigned to each input $a(j)$ $(1 \leq j \leq n)$ of a network. Then there is a unique way to assign a domain value to each output, inlet, and outlet of the network so that the following conditions are satisfied.

(1) The origin and termination of a wire are assigned the same domain values.

(2) If the domain values $w_{j,1}, \ldots, w_{j,q(r(j))}$ are assigned to the inlets $d(j,1), \ldots, d(j,q(r(j)))$ of a gate $h(j)$ $(1 \leq j \leq \ell)$, and if the function f_j is associated with its operation $u(r(j))$, then the domain value $v_j = f_j(w_{j,1}, \ldots, w_{j,q(r(j))})$ is assigned to its outlet $c(j)$.

As a consequence of this, a domain value y_j is assigned to each output $b(j)$ $(1 \leq j \leq m)$ of the network.

THE STANDARD REPRESENTATION

A network can be represented in an obvious way by two files: a logic file and an output file.

The logic file L contains ℓ logic records, one for each gate. The record for $h(j)$ has the format $[c(j), u(r(j)), e(j,1), \ldots, e(j,q(r(j)))]$. The records are in sequence according to $c(j)$.

The output file M contains m output records, one for each output. The output record for $b(j)$ has the format $[b(j), e(j)]$. The records are in sequence according to $b(j)$.

This representation for a network will be called the standard representation. It contains $O(N)$ records, where $N = n+m+\ell$ is a parameter measuring the overall size of the network. If appropriate representations are used for the inputs, outputs, inlets, and outlets, the records will have length $O(\log N)$.

An assignment of values to the inputs of a network can be represented by an input value file X containing n input value records. The record for a(j) has the format $[a(j), x_j]$. The corresponding assignment of values to the outputs can be represented by an output value file Y containing m output value records. The record for b(j) has the format $[b(j), y_j]$. These records are in sequence according to a(j) and b(j), respectively.

The problem of simulating a network in the standard representation can be formulated as follows: given the files L, M, and X, compute Y.

THE ALTERNATE REPRESENTATION

A network can be represented in a different way, also using two files: a gate file and a wire file.

The gate file G contains ℓ gate records, one for each gate. The record for h(j) has the format $[c(j), u(r(j))]$. The records are in sequence according to c(j).

The wire file W contains $q(r(1)) + \ldots + q(r(\ell)) + m$ wire records, one for each wire. The records have the format $[b(j), e(j)]$ or $[d(j,p), e(j,p)]$, and are in sequence according to their terminations (b(j) or d(j,p)).

This representation for a network will be called the alternate representation. It too has $O(N)$ records of length $O(\log N)$.

Let K denote a set of gates. The notation G(K) will denote the subfile of G comprising just those records that represent gates in K, in the same sequence as in G. If H denotes the set of all gates, then $G = G(H)$.

Let I and J denote sets of objects, which may be inputs, outputs, or gates. The notation $W(I \to J)$ will denote the subfile of W comprising just the records representing wires originating in I and terminating in J, in the same sequence as in W. If A denotes the set of all inputs and B the set of all outputs, then $W = W(A \cup H \to H \cup B)$.

During the course of a simulation the values assigned to the origins of various wires will be computed. When this happens, the corresponding wire records will be converted to value records by substituting the assigned values (x_j or v_j) for the origins (a(j) or c(j)) in these records.

The notation V will denote the file containing all the value records corresponding to the wire records in W, in the same sequence as in W. More generally, $V(I \to J)$ will denote the subfile of V containing just the value records corresponding to wire records in $W(I \to J)$, in the same sequence as in $W(I \to J)$. Thus $V = V(A \cup H \to H \cup B)$.

In terms of these notations, the output file M in the standard representation is simply $W(A \cup H \to B)$, and the output value file Y computed by simulating a network in the standard representation is simply $V(A \cup H \to B)$.

The problem of simulating a network in the alternate representation can be formulated as follows: given $G(H)$, $W(H \to H \cup B)$, and $V(A \to H)$, compute $V(H \to B)$. For technical reasons, the records in $W(A \to B)$ and $V(A \to B)$ have been excluded from this definition.

MACHINES

On a machine with random-access storage, the simulation problems formulated above are trivial: one can simulate a network of size N in $O(N)$ operations on records of length $O(\log N)$, or in total time $O(N \log N)$. This is as fast as one can read the representation of the network.

On a machine without random-access storage, these problems are more difficult. We shall show that a machine with one-dimensional tapes can solve them in $O(N \log N)$ operations on records of length $O(\log N)$, or in total time $O(N(\log N)^2)$. This improves previous bounds of $O((N \log N)^2)$ due to Schnorr [1] and $O(N(\log N)^3)$ due to Swamy and the author, independently [unpublished].

We shall need to know that a machine with one-dimensional tapes can perform certain data-processing tasks efficiently.

(1) Given a file containing N records, a machine can classify these records into two subfiles (according to whether or not the value of a certain field exceeds a certain threshold) in $O(N)$ operations on the records.

(2) Given two subfiles which together contain N records and which are both in sequence according to the value of a certain field, a machine can merge these records into a single file (which will also be in sequence according to the value of this field) in $O(N)$ operations on the records.

(3) Given a file containing N records, a machine can sort these records into sequence (according to the value of a certain field) in $O(N \log N)$ operations on the records.

These results come down to us from the days when records were individual punched cards.

We shall be interested in classifying, merging, and sorting records according to the precedence relation among inputs, outputs, inlets, and outlets. If appropriate representations are used for these objects, comparisons for precedence, and more generally operations on records, can be performed in time $O(\log N)$.

As an example, let us show how, given $W(A \to H \cup B)$ and X, a machine can compute $V(A \to H \cup B)$.

(1) Sort the records in $W(A \to H \cup B)$ into sequence according to origin rather than termination. Denote the resulting file $W(A \to H \cup B)^*$.

(2) During a single pass over the files $W(A \to H \cup B)^*$ and X, substitute the values x_i from the input value records $[a(i), x_i]$ for the origins $a(i)$ in the wire records $[b(j), a(i)]$ and $[d(j,p), a(i)]$, obtaining the value records $[b(j), x_i]$ and $[d(j,p), x_i]$. Denote the resulting file $V(A \to H \cup B)^*$.

(3) Sort the records in $V(A \to H \cup B)^*$ back into sequence according to termination. This yields the desired file $V(A \to H \cup B)$.

Since the files $W(A \to H \cup B)$, X, and $V(A \to H \cup B)$ each contain $O(N)$ records, steps (1), (2), and (3) require $O(N \log N)$, $O(N)$, and $O(N \log N)$ operations on records, respectively. This establishes

LEMMA 1: Given $W(A \to H \cup B)$ and X, a machine can compute $V(A \to H \cup B)$ in $O(N \log N)$ operations on records of length $O(\log N)$, or in total time $O(N(\log N)^2)$.

REDUCING THE STANDARD TO THE ALTERNATE REPRESENTATION

The problem of simulating a network in the standard representation can be reduced to that of simulating the network in the alternate representation as follows:

(1) During a single pass over the file L, replace each logic record $[c(j), u(r(j)), e(j,1), \ldots, e(j,q(r(j)))]$ by a gate record $[c(j), u(r(j))]$ and a series of logic records $[d(j,1), e(j,1)]$, $\ldots$, $[d(j,q(r(j))), e(j,q(r(j)))]$. The resulting file of gate records is $G(H)$, and the resulting file of wire records is $W(A \cup H \to H)$.

(2) Merge $W(A \cup H \to H)$ and M (which is in fact $W(A \cup H \to B)$) to obtain $W(A \cup H \to H \cup B)$.

(3) Classify $W(A \cup H \to H \cup B)$ into $W(H \to H \cup B)$ and $W(A \to H \cup B)$.

(4) From $W(A \to H \cup B)$ and X, compute $V(A \to H \cup B)$.

(5) Classify $V(A \to H \cup B)$ into $V(A \to H)$ and $V(A \to B)$.

(6) From $G(H)$, $W(H \rightarrow H \cup B)$, and $V(A \rightarrow H)$, compute $V(H \rightarrow B)$.

(7) Merge $V(A \rightarrow B)$ and $V(H \rightarrow B)$ to obtain $V(A \cup H \rightarrow B)$; this is in fact the desired file Y.

The data flow of this procedure is depicted in Figure 1.

Since all the files dealt with by this procedure contain $O(N)$ records, all the steps except (4) and (6) require $O(N)$ operations on records. It was shown in Lemma 1 above that step (4) requires $O(N \log N)$ operations on records. Step (6) is the problem of simulating the network in the alternate representation. This establishes

LEMMA 2: The problem of simulating a network in the standard representation can be reduced to that of simulating the network in the alternate representation in $O(N \log N)$ operations on records of length $O(\log N)$, or in total time $O(N(\log N)^2)$.

SIMULATION IN THE ALTERNATE REPRESENTATION

To solve the problem of simulating a network in the alternate representation, we shall need to generalize it somewhat.

Let K denote an interval (that is, a contiguous subset) of gates. Let I denote the set of all inputs and gates that precede those in K, and let J denote the set of all outputs and gates that are preceded by those in K.

We shall consider the following problem: given $G(K)$, $W(K \rightarrow K \cup J)$, and $V(I \rightarrow K)$ compute $V(K \rightarrow J)$. This problem clearly subsumes that of simulating a network in the alternate representation, which emerges upon setting $K = H$, $I = A$, and $J = B$.

If K contains only one gate, the problem is trivial. The record in $G(K)$ specifies a function, and the records in $V(I \rightarrow K)$ specify the values of the arguments for which the function is to be evaluated. The records in $W(K \rightarrow K \cup J)$ must in fact be in $W(K \rightarrow J)$, since with only one gate in K there can be no wires that both originate and terminate in KK. When the value of the function is substituted for the origins in these records, the result is the desired file $V(K \rightarrow J)$.

If K contains more than one gate, the problem can be solved recursively. Let K be expressed as the union of two subintervals, $K = I^* \cup J^*$ such that (a) every gate in I^* precedes every gate in J^* and (b) the numbers of gates in I^* and J^* are as nearly equal as possible. Then proceed as follows.

(1) Classify $G(K)$ into $G(I^*)$ and $G(J^*)$.

(2) Classify $W(K \rightarrow K \cup J)$ into $W(I^* \rightarrow K \cup J)$ and $W(J^* \rightarrow K \cup J)$. $W(I^* \rightarrow K \cup J)$ is in fact $W(I^* \rightarrow I^* \cup J^* \cup J)$, and $W(J^* \rightarrow K \cup J)$ is in fact $W(J^* \rightarrow J^* \cup J)$ (since every gate in I^* precedes every gate in J^*, there can be no wires that originate in J^* and terminate in I^*).

(3) Classify $V(I \rightarrow K)$ into $V(I \rightarrow I^*)$ and $V(I \rightarrow J^*)$.

(4) From $G(I^*)$, $W(I^* \rightarrow I^* \cup J^* \cup J)$, and $V(I \rightarrow I^*)$, compute $V(I^* \rightarrow J^* \cup J)$; this is done by recursive application of this procedure, taking I^* for K, I for I, and $J^* \cup J$ for J.

(5) Classify $V(I^* \rightarrow J^* \cup J)$ into $V(I^* \rightarrow J^*)$ and $V(I^* \rightarrow J)$.

(6) Merge $V(I \rightarrow J^*)$ and $V(I^* \rightarrow J^*)$ to obtain $V(I \cup I^* \rightarrow J^*)$.

(7) From $G(J^*)$, $W(J^* \rightarrow J^* \cup J)$, and $V(I \cup I^* \rightarrow J^*)$, compute $V(J^* \rightarrow J)$; this is done by recursive application of this procedure, taking J^* for K, $I \cup I^*$ for I, and J for J.

(8) Merge $V(I^* \rightarrow J)$ and $V(J^* \rightarrow J)$ to obtain $V(K \rightarrow J)$.

The data flow of this procedure is depicted in Figure 2.

The invocations of this procedure correspond to the nodes of a tree. The initial invocation corresponds to the root. The recursive invocations by the initial invocation correspond to the successors of the root, and so forth to the final invocations, which correspond to the leaves. Since the number of gates to be simulated is divided as evenly as possible at each invocation, the height of this tree is $O(\log N)$.

Let us consider the time spent in the final invocations. Each such invocation is responsible for the simulation of one gate. A gate record is processed by only one such invocation, the one responsible for simulating the gate it represents. A wire record or value record is processed by at most two such invocations, the ones responsible for simulating the gates at which the wire it represents originates or terminates. The total number of gate, wire, and value records is $O(N)$, and each of them is processed $O(1)$ times, so the total number of records processed by all the final invocations is $O(N)$.

Let us now consider the time spent in the classify and merge steps of non-final invocations. A gate record is classified only during non-final invocations corresponding to nodes on the path from the root to a leaf, this leaf being the one that corresponds to the final invocation responsible for simulating the gate represented by the record. There are $O(\log N)$ such non-final invocations. A wire or value record is classified or merged only during non-final invocations corresponding to nodes on the paths from the root to at most two leaves, these leaves being those that correspond to the final invocations responsible for simulating the gates at which the wire represented by the record originates or terminates. There are $O(\log N)$ such non-final invocations, and each of them classifies or merges a record $O(1)$ times, since there are no cycles in the data flow of the procedure. The total number of gate, wire, and value records is $O(N)$, and each of them is classified or merged $O(\log N)$ times, so the total number of records processed by classify and merge steps in all non-final invocations is $O(N \log N)$.

This establishes

LEMMA 3: The problem of simulating a network in the alternate representation can be solved in $O(N \log N)$ operations on records of length $O(\log N)$, or in total time $O(N(\log N)^2)$.

In view of Lemma 2, this in turn establishes

THEOREM: The problem of simulating a network in the standard representation can be solved in $O(N \log N)$ operations on records of length $O(\log N)$, or in total time $O(N(\log N)^2)$.

ADDENDUM

Michael Fischer and Michael Paterson have obtained a very general result from which the result of the present paper can be deduced very easily. They show that an "event list" or "mailbag" can be maintained efficiently without random-access storage. This result, and other applications of it, will be reported elsewhere [2].

REFERENCES

[1] C.-P. Schnorr, "The Network Complexity and the Turing Machine Complexity of Finite Functions," Acta Informatica, V. 7 (1976) pp. 95-107.

[2] M. J. Fischer, M. S. Paterson and N. J. Pippenger, (in preparation).

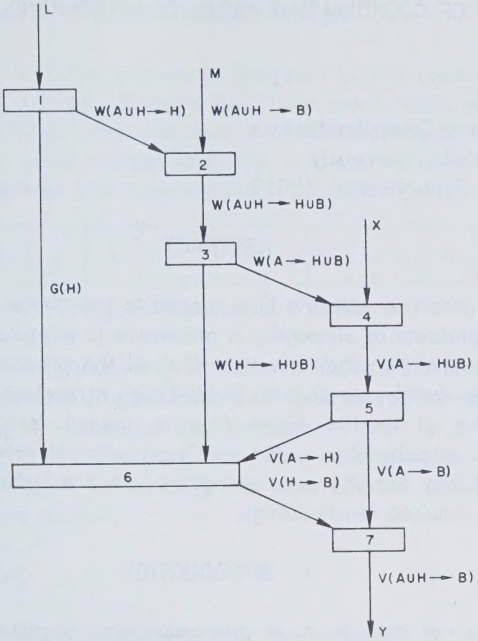

L
M
1
W(AUH → H)
W(AUH → B)
2
W(AUH → HUB)
X
3
W(A → HUB)
G(H)
4
W(H → HUB)
V(A → HUB)
5
V(A → H)
V(H → B)
V(A → B)
6
7
V(AUH → B)
Y

Figure 1. The data flow of the reduction to the alternate representation.

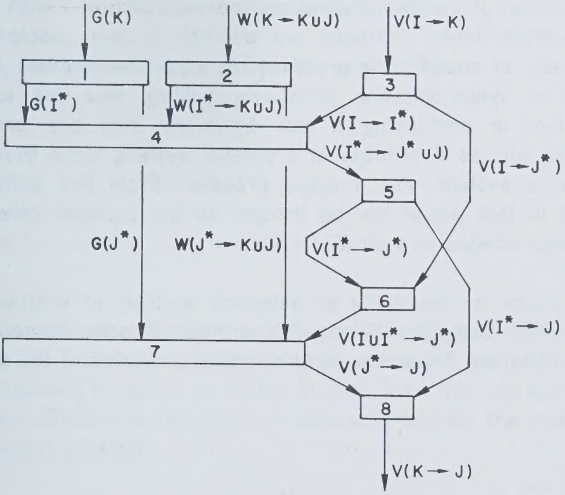

G(K)
W(K → KUJ)
V(I → K)
1
2
3
G(I*)
W(I* → KUJ)
V(I → I*)
4
V(I* → J* UJ)
V(I → J*)
5
G(J*)
W(J* → KUJ)
V(I* → J*)
6
V(I* → J)
7
V(IUI* → J*)
V(J* → J)
8
V(K → J)

Figure 2. The data flow of the simulation in the alternate representation.

THE COMPLEXITY OF COORDINATING PARALLEL ASYNCHRONOUS PROCESSES

H. T. KUNG
Department of Computer Science
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

ABSTRACT

This paper is concerned with the time needed to coordinate parallel asynchronous processes. The problem of scheduling k processes to execute n independent tasks is considered. It is shown that to ensure that all the processes are always doing useful work, in the worst case at least $((k/2)-1)\log_2 n$ reschedulings are needed and consequently order at least k log n time is needed for process coordination. Moreover, a class of schedules, called the "equal interval schedules" in the paper, are proposed and they are shown to be optimal within a factor of two with respect to the number of required reschedulings.

1. INTRODUCTION

When a number of asynchronous processes work together to solve a single problem, process coordinations are usually required. In the programming systems area, various synchronization concepts for ensuring correct process communication have been proposed (see, e.g., [1] for an overview). In asynchronous multiprocessor algorithms area, analytic techniques have been proposed to analyze the expected computation time (see, e.g., [2,3,4,5]). In this paper, we are not concerned with forms of synchronization, nor are we concerned with the analysis of expected computation times. Instead, we address a new problem, namely, the inherent complexity of coordinating processes in a parallel system. We regard the execution time of synchronization primitives, waiting time for entering critical sections, time lost in performing a task by more than one process, process rescheduling time, etc. as overheads in a parallel system, since these times would not be spent in a system with a single process. From this point of view the problem studied in this paper can be thought as the problem of determining the minimum overheads needed in a parallel system.

The present paper is essentially an extended abstract of another paper which appears as a CMU Computer Science Department Report, October 1977. The reader is referred to the Report for more complete treatment of the subject.

This research was supported in part by the National Science Foundation under Grant MCS75-222-55 and the Office of Naval Research under Contract N00014-76-C-0370, NR044-422.

2. THE MAIDS PROBLEM

It is instructive to look the problem investigated in this paper as a problem of coordinating maids of a hotel. Suppose that there are k maids who simultaneously clean n rooms of a hotel. For notational convenience, throughout the paper we assume that k is an even integer and n is a power of two. The rooms are numbered by integers from 1 to n, as depicted in Fig. 1.

Fig. 1: n rooms of a hotel.

A maid can be assigned to clean any of the n rooms, but the time to clean a room by a maid is unpredictable (for reasons such as that rooms may be in different conditions before they are cleaned). For efficiency reasons, it is desirable to have all the maids working all the time. Informally, the maids problem is to minimize the total coordination time spent for ensuring that none of the k maids are idle until all the n rooms have been cleaned.

The One-Maid Schedule

The problem is trivial when there is only one maid. An obvious schedule is to let the maid clean the rooms one by one, say, in the increasing order of room numbers (cf. Fig. 2).

Fig. 2. A schedule for one maid.

More precisely, the maid executes the following procedure:

```
i ← 1;
while i < n + 1
  do
    clean room i;
    i ← i + 1
  od
```

Note that besides cleaning rooms, the maid also has to deal with the bookkeeping problem, i.e., to increment the pointer i and to check the loop exit condition. It is clear that one cannot hope to do better in time than this obvious schedule. We assume that this schedule is the minimum time schedule for the one maid case, and call it *the one-maid schedule*.

A Naive Two-Maid Schedule

Suppose that there are two maids for cleaning n rooms. A naive schedule is as follows. Initially let maid 1 clean rooms 1 to n/2 and maid 2 clean rooms (n/2)+1 to n. Due to fluctuations in cleaning times, maid 1, say, finishes cleaning all her rooms before maid 2. Maid 1 has to be rescheduled at this point to help cleaning those rooms which have not been cleaned by maid 2, for otherwise maid 1 would become idle while maid 2 is still working on her rooms. The rescheduling policy is that maid 1 starts cleaning the second half of the remaining uncleaned rooms left by maid 2 in the increasing order of room numbers. Since it may happen that when one maid finishes cleaning all the rooms assigned to her the other maid has not finished cleaning even a single room, the number of reschedulings can be as large as $\log_2 n$.

A Formal Statement of the Maids Problem

A _schedule_ of a maid is a sequence of integers drawn from integers from 1 to n. Schedules of all the k maids constitute a _scheduling configuration_. _The matrix corresponding to a scheduling configuration_ is the matrix whose ith row is the schedule of maid i in the scheduling configuration. A _k-maid schedule_ consists of an initial scheduling configuration for k maids and a _rescheduling policy_. The rescheduling policy defines a _rescheduling condition_ for each maid such that when the condition becomes true the corresponding maid may obtain a new schedule and change the rescheduling conditions of one or more maids. We often say that a _rescheduling occurs_ when a rescheduling condition becomes true. The rescheduling policy also specifies how the new schedule is derived and how some rescheduling conditions are changed when a rescheduling occurs. Under a k-maid schedule, each of the k maids cleans rooms in the order specified by her current schedule until her rescheduling condition becomes true, and then after obtaining a new schedule she starts cleaning rooms according to the new schedule. In the following we illustrate these definitions by considering the two schedules discussed above.

For the one-maid schedule, the initial scheduling configuration consists of just one schedule, which is the sequence 1,2,...,n, and the rescheduling policy is that when $i \geq n+1$ the maid obtains the empty sequence as her new schedule. For the naive two-maid schedule, the initial scheduling configuration consists of two sequences 1,2,...,n/2 and (n/2)+1,...,n, and the rescheduling policy is that when a maid realizes that the room she is scheduled to clean next has already been cleaned by the other maid she will get a new schedule which is the second half of the remaining schedule of the other maid.

A k-maid schedule is called an _adaptive schedule_ if the schedule guarantees that none of the k maids will become idle until all the n rooms have been cleaned. It should be clear that in an adaptive schedule some extra time must be spent in coordinating maids. More precisely, we define the _coordination time_ of an adaptive k-maid schedule to be the difference between the total time spent by the k maids under the schedule and the time taken by the one-maid schedule. It is easy to see that for the naive two-maid schedule, the coordination time is order $\log_2 n$ in the worst case since as many as $\log_2 n$ reschedulings can occur. The maids problem is to find adaptive k-maid schedules which have the minimum worst case coordination time. We shall use $C_k(n)$ to denote the worst case coordination time of an adaptive k-maid schedule. We are also interested in the number of reschedulings occurred in an adaptive schedule, since it is closely related to the coordination time. We shall use $R_k(n)$ to denote the maximum number of reschedulings which may occur in an adaptive k-maid schedule. In the next two sections, we shall mostly deal with $R_k(n)$ rather than $C_k(n)$.

3. THE INTERVAL SCHEDULES

In this section we propose a class of adaptive k-maid schedules, called the "interval schedules", which correspond in a natural way to partitions of an interval into k/2 subintervals. We show that "equal interval schedules" are optimal among all "interval schedules".

Equal Interval Schedules

The two-maid interval schedule. The idea is to let maid 1 start working from room 1 in the increasing order of room numbers and in the mean time let maid 2 start working from room n in the decreasing order of room numbers. As depicted in Fig. 3, the schedule corresponds naturally to an interval of length n with two arrows pointing to each other at the endpoints of the interval.

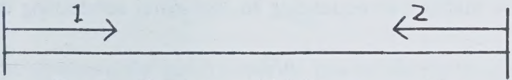


Fig. 3. The two-maid interval schedule.

More precisely, maid 1 executes the procedure:

```
i ← 1;
while i < j
  do
    clean room i;
    i ← i + 1
  od
clean room i
```

and maid 2 executes the procedure:

```
j ← n;
while i < j
  do
    clean room j;
    j ← j - 1
```

od

In other words, the matrix corresponding to the initial scheduling configuration is

$$\begin{matrix} 1 & 2 & 3 & \ldots & n \\ n & n-1 & n-2 & \ldots & 1 \end{matrix}$$

The rescheduling policy is that if maid 1 finds $i \geq j$, she cleans room i and then stops working, while if maid 2 finds $i \geq j$, she stops working immediately. Note that it is possible that the room which is finished last is cleaned by both maids. Besides the time spent for this possible redundant work, the total time spent by the two maids, including both the time to clean rooms and to perform the bookkeeping operations, is exactly the same as the time taken by the one-maid schedule. Since the redundancy involves cleaning one room and performing a constant number of bookkeeping operations, we assume that it takes a constant time. This implies that the present two-maid schedule has a constant coordination time, while the naive two-maid schedule given earlier needs $O(\log_2 n)$ coordination time. (The two-maid interval schedule corresponds to the use of a deque. The advantage of using a deque rather than a stack or queue for adaptive parallel systems was pointed out before in [3, Section 5], and has been shown in applications (see, e.g., [4]).)

The four-maid equal interval schedule. The idea of the two-maid interval schedule generalizes to a four-maid equal interval schedule. Initially, we partition the interval of length n into two subintervals of equal length and we assign two maids to work on each of the two subintervals in opposite directions (see Fig. 4). In other words, the matrix corresponding to the initial scheduling configuration is:

$$\begin{matrix} 1 & 2 & \ldots & n/2 \\ n/2 & (n/2)-1 & \ldots & 1 \\ (n/2)+1 & (n/2)+2 & \ldots & n \\ n & n-1 & \ldots & (n/2)+1 \end{matrix}$$

When any of the two subintervals is finished, the corresponding two maids find the middle point of the unfinished portion of the other subinterval and start working from the middle point in opposite directions. This rescheduling policy is illustrated in Fig. 5.

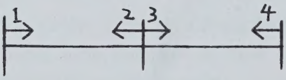


Fig. 4. The initial scheduling configuration of the four-maid equal interval schedule.

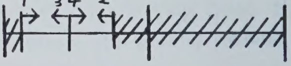


Fig. 5. The rescheduling policy of the four-maid equal interval schedule. The shaded regions represent rooms which have already been cleaned.

It should be clear that under this schedule when two maids meet at least half of the rooms have already been cleaned. Hence after two reschedulings, one for each of the two maids, the number of uncleaned rooms is reduced at least by a factor of two. Therefore, we have shown that

$$R_4(n) = 2 \log_2 n \,.$$

The six-maid equal interval schedule. The initial scheduling configuration is depicted in Fig. 6.

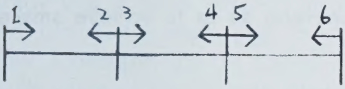


Fig. 6. The initial scheduling configuration of the six-maids equal interval schedule.

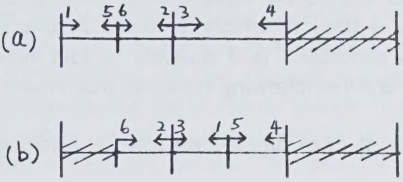


Fig. 7. (a) A scheduling configuration after one interval is finished.
(b) A scheduling configuration after two intervals are finished.

The rescheduling policy is that one of the largest unfinished intervals should be equally split. This is illustrated in Fig. 7. It is easy to check that after two intervals are finished, i.e., after four reschedulings, the original interval size is reduced at least by a factor of two. This implies that

$$R_6(n) = 4 \log_2 n \,.$$

The result can be immediately generalized to the following theorem:

Theorem 1: For the k-maid equal interval schedule, $R_k(n) = (k-2) \log_2 n$.

Since the time taken by each rescheduling in the k-maid equal interval schedule is $O(k)$, we have the following corollary:

Corollary1: For the k-maid equal interval schedule, $C_k(n) = O(k^2 \log n)$.

Optimality of Equal Interval Schedules among all Interval Schedules

K-maid equal interval schedules are special cases of k-maid interval schedules where intervals are not necessarily equally partitioned in the initial scheduling configuration or in the subsequent reschedulings. Fig. 8, for example, gives the initial scheduling configuration of a six-maid interval schedule which is not the six-maid equal interval schedule.

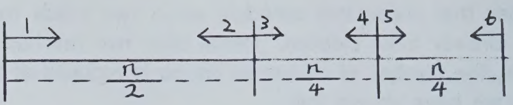


Fig. 8. The initial scheduling configuration of the six-maid interval schedule which is not the six-maid equal interval schedule.

One can check that if the rescheduling policy is that one of the largest unfinished intervals should be equally split, then for this six-maid interval schedule we have $R_6(n) = 4 \log_2 n$, which, as noted before, is also achieved by the six-maid equal interval schedule. Is it possible to split intervals unevenly so as to achieve smaller $R_k(n)$? By the following theorem, the answer is "no".

Theorem 2: For any k-maid interval schedule, $R_k(n) \geq (k-2) \log_2 n$.

Proof

Consider any k-maid interval schedule. The proof is based on an adversary argument using the following strategies:

(i) The first finishing interval is the smallest one among the k/2 subintervals in the initial configuration.

(ii) After a rescheduling, the smaller of the two subintervals resulting from the partition corresponding the rescheduling is the subinterval which will finish next.

It can be seen that after (k/2)-1 intervals are finished, there is a one-to-one correspondence between finished and unfinished intervals such that an unfinished interval is always no shorter than the corresponding finished interval. This implies that after (k/2)-1 intervals are finished, i.e., after k-2 reschedulings, the original interval size is reduced at most by a factor of two. The proof of the theorem is complete. By Theorems 1 and 2, we have the following corollary:

Corollary 2: The k-maid equal interval schedule is optimal with respect to the number of required reschedulings, among all k-maid interval schedules.

4. LOWER BOUNDS

In the preceding section we have shown that equal interval schedules are optimal with respect to the number of required reschedulings among all interval schedules. This optimality result is extended in this section to a much wider class of adaptive schedules.

We say that a rescheduling condition is minimum if a maid will obtain a new schedule only when the room she is scheduled to clean next has already been cleaned by herself or another maid, or when she has cleaned all the rooms specified by her current schedule. In the former case, the maids involved, who may be the same maid, are referred to as the maids corresponding the rescheduling. Observe that interval schedules are schedules with the minimum rescheduling condition. In this section we consider adaptive schedules with the minimum rescheduling condition. The assumption that schedules use the minimum rescheduling condition is a very mild restriction; it seems that the class of such schedules is large enough to contain all interesting adaptive schedules.

Consider a k-maid schedule with the minimum rescheduling condition. We shall show that at least $((k/2)-1)\log_2 n$ reschedulings are needed in the worst case. In the proof, we allow both the maids corresponding to one rescheduling to obtain new schedules; this of course only strengthens the lower bound result. Note that each rescheduling is associated with a number, which is the number of rooms that have been cleaned at the time when the rescheduling occurs. We assume that the first rescheduling occurs at a time when a total of x_1 rooms have been cleaned and that x_1 is not larger than the number associated with any other possible first rescheduling.

Lemma 1: $x_1 \le 2n/k$.

Proof

Consider the k by n matrix corresponding to the initial scheduling configuration of the k-maid schedule. Let A be the submatrix consisting of the first $x_1/2$ columns of the original k by n matrix. A must be a matrix with no empty entries and numbers in A must be all distinct, for otherwise the first rescheduling might occur at a time when fewer than x_1 rooms have been cleaned. Since there are $kx_1/2$ elements in the submatrix A, we must have $n \ge kx_1/2$. This implies $x_1 \le 2n/k$.

Theorem 3: For any k-maid schedule with the minimum rescheduling condition, we have

$$R_k(n) \ge (k/2-1)\log_2 n,$$

even if it is allowed that both the maids corresponding to one rescheduling may obtain new schedules.

Proof.

Consider the case when k = 6. Let x_1 be defined as above. Let x_2 be the number of additional rooms which will have been cleaned at the time when the second rescheduling occurs. It suffices to show that $x_1 + x_2 \le n/2$. Without loss of generality, let maids 1 and 2 be the ones corresponding to the first rescheduling. Consider the scheduling configuration right after the first rescheduling is complete. The matrix corresponding the scheduling configuration appears in Fig. 9.

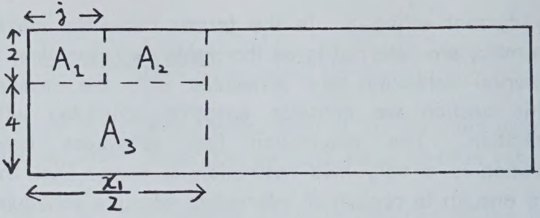


Fig. 9. The scheduling configuration after the first rescheduling is complete.

Case 1. Suppose that $A_1 \cup A_2 \cup A_3$ contains some empty entry or the number of some room which has already been cleaned. Then the second scheduling may occur with $x_2 \leq x_1/2$. By Lemma 1, it follows that $x_1 + x_2 \leq 3x_1/2 \leq n/2$.

Case 2. Suppose that $A_1 \cup A_2 \cup A_3$ does not contain any empty entry, nor does it contain the number of any room which has been cleaned. Note that by the proof of Lemma 1 numbers in A_3 are distinct. Let m be the largest value of j such that numbers in A_1 are distinct and that none of them appear in A_3 (cf. Fig. 9). Then

$$2m + 4(x_1/2) + x_1 \leq n,$$

since numbers in $A_1 \cup A_3$ are all distinct, and the second scheduling may occur with

$$x_2 \leq m + x_1/2.$$

The above two inequalities imply that $x_1 + x_2 \leq n/2$. We have shown that Theorem 3 holds for k = 6. It is not difficult to see that the same technique can be used to prove the theorem for general k.

By Theorems 1 and 3 we have the following result:

Corollary 3: The k-maid equal interval schedule is optimal within a factor of two with respect to the number of required reschedulings, among all k-maid schedules with the minimum rescheduling condition.

Strictly speaking, at the time when a maid realizes that she has to be rescheduled, she has already done some operation which would not have to be performed under the one-maid schedule (namely, the test operation through which she can tell whether or not she has to be rescheduled). Hence by Theorem 3, we have the following corollary:

Corollary 4: For any k-maid schedule with the minimum rescheduling condition, the coordination time is order at least k log n, even if we assume that all the reschedulings take no time.

On the other hand, suppose that a rescheduling takes order at least k time as in the k-maid equal interval schedule. Then by Theorem 3 for any k-maid schedule with the minimum rescheduling condition the coordination time is order at least k^2 log n. This together with Corollary 1 implies that the k-maid equal interval schedule is optimal in order of magnitude with respect to the coordination time.

5. CONCLUSION

The concept of coordination time in a parallel asynchronous system has been formulated. It has been shown through the maids problem that the coordination time required for k processes to execute n independent tasks is order at least k log n in the worst case if the minimum rescheduling condition is used. As a consequence, equal interval schedules are shown to be essentially optimal with respect to all schedules with the minimum rescheduling condition.

REFERENCES

1. Andler, S., Synchronization Primitives and the Verification of Concurrent Programs, Department of Computer Science Report, Carnegie-Mellon University, May 1977.

2. Baudet, G., Brent, R. P. and Kung, H. T., Parallel Execution of a Sequence of Tasks on an Asynchronous Multiprocessor, Department of Computer Science Report, Carnegie-Mellon University, June 1977.

3. Kung, H. T., Synchronized and Asynchronous Parallel Algorithms for Multiprocessors, Algorithms and Complexity: New Directions and Recent Results, edited by J. F. Traub, Academic Press, 1976, 153-200.

4. Kung, H. T. and S. W. Song, An Efficient Parallel Garbage Collection System and Its Correctness Proof, Computer Science Department Report, Carnegie-Mellon University, September 1977. Also in Proc. of the 18th Annual IEEE Symposium on Foundations of Computer Science, October 1977.

5. Robinson, J. T., Analysis of Asynchronous Multiprocessor Algorithms with Application to Sorting, Computer Science Department Report, Carnegie-Mellon University, June 1977. Also to appear in Proc. 1977 International Conference on Parallel Processing, August 1977.

PROBABILITIES OF LARGE DERIVATIONS FOR SOME STOCHASTIC EQUATIONS AND APPLICATIONS

GEORGE C. PAPANICOLAOU
Courant Institute, New York University, New York, N. Y. 10012

INTRODUCTION

In dealing with stochastic equations modeling fluctuation phenomena it is convenient to single out a small parameter ε, the ratio of the correlation time of the fluctuating coefficients to some typical time scale of the system as a whole, and carry out an asymptotic analysis as $\varepsilon \to 0$. There are many different ways the analysis can be implemented depending on the type of questions one wishes to focus attention on. We discuss here briefly some recent work that focuses on large deviation phenomena [1]-[5].

1. AN EXAMPLE

Suppose that $y(t)$, $t \geq 0$, is the random telegraph process i.e. a continuous time Markov process taking the values α and β and staying in each state for an exponentially distributed length of time, with parameters q_α, q_β. The infinitesimal generator of $y(t)$ is given by

$$\frac{d}{dt} P\left\{ y(t) = \begin{matrix}\alpha\\ \beta\end{matrix} \;\middle|\; y(0) = \begin{matrix}\alpha\\ \beta\end{matrix} \right\}\Bigg|_{t=0} \equiv Q$$

with Q the 2×2 matrix

$$Q = \begin{pmatrix} -q_\alpha & q_\alpha \\ q_\beta & -q_\beta \end{pmatrix} \tag{1}$$

The process $y(t)$ is ergodic and

$$\lim_{t\uparrow\infty} P\left\{ y(t) = \begin{matrix}\alpha\\ \beta\end{matrix} \;\middle|\; y(0) = y \right\} = \frac{1}{q_\alpha + q_\beta} \begin{pmatrix} q_\beta \\ q_\alpha \end{pmatrix} \tag{2}$$

independently of the starting point y ($= \alpha$ or β).

Let $F(x,\alpha)$ and $F(x,\beta)$ be two R^n-valued smooth functions of $x \in R^n$. Consider the stochastic differential equation[†]

$$\frac{dx^\varepsilon(t)}{dt} = F(x^\varepsilon(t), y^\varepsilon(t)) \;, \quad x^\varepsilon(0) = x \;, \quad y^\varepsilon(t) \equiv y(t/\varepsilon) . \tag{3}$$

It is easily seen that the joint process $\{x^\varepsilon(t), y^\varepsilon(t);\ t \geq 0\}$ is a Markov process with state space $R^n \times \{\alpha,\beta\}$.

[†] Here $y^\varepsilon(0) = y$.

We observe that the small parameter ε enters in (3) in a manner consistent with the interpretation in the introduction. To fully articulate this point, however we must establish the proper relationship between (3) and its averaged or deterministic counterpart

$$\frac{d\bar{x}(t)}{dt} = \bar{F}(\bar{x}(t)) , \quad \bar{x}(0) = x. \tag{4}$$

Here

$$\bar{F}(x) = \frac{1}{q_\alpha + q_\beta} [q_\beta F(x,\alpha) + q_\alpha F(x,\beta)] . \tag{5}$$

This is the law of large numbers for stochastic equations like (3): for any $T < \infty$ and all $\delta > 0$,

$$P\left\{ \sup_{0 \leq t \leq T} |x^\varepsilon(t) - \bar{x}(t)| > \delta \right\} \to 0 , \quad \text{as} \quad \varepsilon \to 0 . \tag{6}$$

It is not hard to prove (6), under very general conditions in fact and in a variety of ways; one proof[†] is very similar to the proof of the validity of the averaging method for deterministic systems [7]. This is more than an analogy since in fact

$$\bar{F}(x) = \lim_{T \uparrow \infty} \frac{1}{T} \int_0^T F(x,y(s))\, ds \quad \text{with probability 1.}$$

Now suppose A is a compact subset of R^n and that $\bar{x}(t) \notin A$. Consider

$$u_\varepsilon(x,y,t;A) \equiv E\left\{ x^\varepsilon(t) \in A \mid x^\varepsilon(0) = x,\ y^\varepsilon(0) = y \right\} , \quad t \geq 0 . \tag{7}$$

It is intuitively clear that because of (6) $u_\varepsilon \to 0$ as $\varepsilon \to 0$ for each x,y and $t > 0$ fixed. It turns out that, without any additional restrictions,[††]

$$\lim_{\varepsilon \downarrow 0} \varepsilon \log u_\varepsilon(x,y,t;A) = - S(x,t;A) \tag{8}$$

where $S > 0$ is computed in the next section as the solution of a calculus of variations problem. If $\bar{x}(t) \in A$ then $S(x,t;A) = 0$ and (8) has no content.

[†] Cf. the surveys [6] for a proof amd many references to related work.

[††] In fact for much more general systems than (3) (cf. [4]). A much more comprehensive statement than (8) is also known [4].

The result (8) is the solution of the large deviations question. In words it says this: given that the solution of (3) satisfies the law of large numbers, i.e. (6), the probability that $x^{\varepsilon}(t)$ will be found in A away from $\bar{x}(t)$ goes to zero exponentially as $\varepsilon \to 0$ and S is the decay rate.

2. THE EXPANSION

If the boundary of A is smooth and t is sufficiently small, what follows is perfectly rigorous with only routine details left out (and a lot of information beyond (8) can be obtained in principle). The significance of (8) lies, however, in its being a globl (in t) statement. The full proof requires new and far reaching techniques that are different from the so-called WKB method that follows (as a rule in general, WKB methods will nevertheless produce correct answers).

We note that u_{ε} satisfies the equation

$$\begin{aligned} \frac{\partial u_{\varepsilon}}{\partial t} &= \frac{1}{\varepsilon} Q u_{\varepsilon} + F(x,y) \cdot \frac{\partial u_{\varepsilon}}{\partial x}, \qquad t > 0, \\ u_{\varepsilon}(x,y,0;A) &= \chi_A(x,y) \end{aligned} \tag{9}$$

Here $\chi_A(x,y)$ is the indicator function of the set $A \subset R^n$ regarded as a subset of $R^n \times S$, $S = \{\alpha,\beta\}$. Also u_{ε} is regarded as a 2-vector, since $y \in S$ is α or β, and Q acts on u as a matrix (cf. (1)). To analyze (9) as $\varepsilon \to 0$ we first replace $\chi_A(x,y)$ by $\exp(-\tilde{S}(x)/\varepsilon)$ where

$$\begin{aligned} &\tilde{S}(x) \text{ is smooth}, \quad \tilde{S}(x) \equiv 0, \quad x \in A \quad \text{and} \\ &\tilde{S}(x) \to \infty \quad \text{as} \quad |x| \to \infty \end{aligned} \tag{10}$$

By choosing $\tilde{S}$'s suitably we can sandwich u_{ε} with $\tilde{u}_{\varepsilon}$'s which are solutions of (9) with $u_{\varepsilon}(t=0) = \exp(-\tilde{S}/\varepsilon)$. It suffices to expand $\tilde{u}_{\varepsilon}$, therefore, and then in the end let $\tilde{S}(x)$ go to infinity for all $x \notin A$.

The expansion for $\tilde{u}_{\varepsilon}$ is taken in the form

$$\tilde{u}_{\varepsilon} = e^{-\bar{S}(x,t)/\varepsilon}\left[u_0(x,y,t) + \varepsilon u_1(x,y,t) + \dots\right] \tag{11}$$

with $\bar{S}(x,0) = \tilde{S}(x)$, and u_0 and u_1 appropriately chosen at $t = 0$. Inserting (11) into (9) and equating to zero coefficients of equal powers of ε we find that

$$[Q - F(x,y) \cdot \frac{\partial \bar{S}(x,t)}{\partial x} + \frac{\partial \bar{S}(x,t)}{\partial t}]u_0 = 1 \tag{12}$$

$$[Q - F(x,y) \cdot \frac{\partial \bar{S}(x,t)}{\partial x} + \frac{\partial \bar{S}(x,t)}{\partial t}]u_1 + F(x,y) \cdot \frac{\partial u_0}{\partial x} - \frac{\partial u_0}{\partial t} = 0, \text{ etc.} \qquad (13)$$

First we consider (12). Put for the moment

$$k = \frac{\partial \bar{S}}{\partial x} \quad \text{(an n-vector)} \ , \qquad \omega = - \frac{\partial \bar{S}}{\partial t} \qquad (14)$$

and with (k,x) fixed consider the eigenvalue problem

$$(Q-k \cdot F(x,y))\phi(y;k,x) = \omega(k,x)\phi(y;,,x) \qquad (15)$$

with $y \in S = \{\alpha,\beta\}$. Of course (15) is the 2×2 matrix eigenvalue problem

$$\begin{pmatrix} -q_\alpha - k \cdot F(x,\alpha) - \omega & q_\alpha \\ q_\beta & -q_\beta - k \cdot F(x,\beta) - \omega \end{pmatrix} \begin{pmatrix} \phi(\alpha) \\ \phi(\beta) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} , \qquad (16)$$

But it is useful to keep the general notation which shows better what is involved.

We know that (15) always has a maximal eigenvalue $\omega(k,x)$ to which there corresponds a positive right eigenvector $\phi(y;k,x)$ and a positive left eigenvector[†] $\psi(y;k,x)$ (positive for each $y \in S$). Moreover the eigenvalue $\omega(k,x)$ is isolated (for (16) this is obvious; more generally it is true if Q is ergodic in a suitable sense), differentiable in (k,x) (so are the eigenvectors) and convex in k i.e., the matrix $(\partial^2\omega/\partial k_i \partial k_j)$ is positive semi-definite for each (k,x). We shall assume that it is positive definite. Note further that

$$\omega(0,x) \equiv 0 \ , \qquad \phi(y;0,x) \equiv 1 \ , \qquad \psi(y;0,x) = \bar{P}(y;x) \qquad (17)$$

where $\bar{p}$ is the invariant measure of Q and 1 stands for the vector all of whose components equal one.

Now we choose $\bar{S}(x,t)$ to satisfy the nonlinear Hamilton-Jacobi equation[*]

$$\bar{S}_t + \omega(\nabla\bar{S},x) = 0 \ , \qquad t > 0 \ , \qquad \bar{S}(x,0) = \tilde{S}(x) \ . \qquad (18)$$

[†] Normalized so that $\sum_{y \in S} \phi(y;k,x)\psi(y;k,x) \equiv (\psi,\phi) = 1$.

[*] $\nabla S = \partial S/\partial x$, $S_t = \partial S/\partial t$.

If $\tilde{S}(x)$ is convex* and smooth, (18) will have a smooth solution globally in time. Without convexity the solution will exist up to some time and not later. As long as $\bar{S}(x,t)$ is smooth u_0 , u_1 , etc. can be computed now as usual. Namely

$$u_0(x,y,t) = \bar{u}_0(x,t)\phi(y,\nabla\bar{S},x) \tag{19}$$

and $\bar{u}_0(x,t)$ satisfies a first order partial differential equation (a transport equation) obtained from the solvability condition (Fredholm alternative) for (13). We focus attention on $\bar{S}$ called the action function.

3. THE ACTION FUNCTION

Let us look at (18) in some detail. Since ω is convex in k, the Lagrargian function

$$L(x,\dot{x}) = \sup_k (\dot{x}\cdot k - \omega(k,x)) \tag{20}$$

is well defined and the sup is achieved at the k obtained by solving $\dot{x} = \partial\omega/\partial k$. The Lagrangian is strictly convex in $\dot{x}$ and positive except at $\dot{x} = -\bar{F}$ (cf. (5)) where it vanishes. This is seen as follows.

Differentiating (15) once with respect to k we obtain

$$\left(Q - k \cdot F - \omega\right) \frac{\partial\phi}{\partial k_i} = \left(F_i + \frac{\partial\omega}{\partial k_i}\right)\phi \tag{21}$$

and hence

$$\frac{\partial\omega}{\partial k_i}(k,x) = -(\psi(\cdot,k,x), F_i(x,\cdot)\phi(\cdot,k,x)), \; i = 1,2,\dots,n. \tag{22}$$

$$\frac{\partial\phi}{\partial k_i} = \left(Q - k \cdot F - \omega\right)^{-1}\left(F_i + \frac{\partial\omega}{\partial k_i}\right)\phi \tag{23}$$

From (22) it follows that when $\dot{x} = -\bar{F} = \partial\omega/\partial k$, necessarily $\psi = \bar{p}$ and $\phi = 1$ i.e. $k = 0$. Hence $L(x,-\bar{F}) \equiv 0$ by (17). The positivity of L for $\dot{x} \neq -\bar{F}$ follows from the fact that L is convex in $\dot{x}$ and $k = \partial L/\partial\dot{x}$ is zero only when $\dot{x} = -\bar{F}$ i.e. the minimum value is zero and is taken at $\dot{x} = -\bar{F}$.

Consider Hamilton's equations

$$\begin{aligned} \dot{x}(t) &= \frac{\partial\omega}{\partial k}(x(t),k(t)), \qquad x(0) = x \\ \dot{k}(t) &= -\frac{\partial\omega}{\partial x}(x(t),k(t)), \qquad k(0) = \tilde{S}(x) \end{aligned} \tag{24}$$

---

* That is A is convex and $\tilde{S}(x)$ outside A is convex.

which have global solutions. From (18) we obtain that

$$\bar{S}\big(x(t),t\big) = \tilde{S}(x) + \int_0^t L\big(x(s),\dot{x}(s)\big)\,ds \tag{25}$$

If the mapping $x \to x(t;x)$ $(x(0;x) = x)$ is one to one for $0 \le t < t_0$ then every point (x,t), $0 < t < t_0$, can be reached by a path of (24) with suitable starting point. Otherwise conjugate or caustic points appear. Clearly

$$\bar{S}(x,t) = \inf_{\xi(t)=x}\left[\int_0^t L(\xi(s),\dot{\xi}(s))\,ds + \tilde{S}\big(\xi(0)\big)\right] \tag{26}$$

and, again, this will be smooth away from caustic points and will satisfy (18). However, (26) is a global statement (in t).

Now recall that $\tilde{S}(x)$ was introduced as a smoothing operator. We let $\underline{S}(x)$ tend to the singular function

$$\tilde{S}(x) = \begin{cases} 0 & x \in A \\ +\infty & x \notin A \end{cases} \tag{27}$$

Now the expansion (11) breaks down at $t = 0$ and on the "shadow boundary" as we describe below. The $S(x,t)$ that appears in (8) can now be seen to be

$$S(x,t) = \inf_{\substack{\xi(t)=x \\ \xi(0)\in A}}\left[\int_0^t L\big(\xi(s),\dot{\xi}(s)\big)\,ds\right] \tag{28}$$

and in fact $\xi(0)$ must be on ∂A. One may call the passage from (26) to (28), removing of the penalty function ($\tilde{S}$ = penalty).

Regarding (28) we observe the following. Suppose first that, given (x,t), $t > 0$ there is an $x_0 \in A$ such that if

$$\frac{d\bar{x}(t)}{dt} = \bar{F}\big(\bar{x}(s)\big), \qquad \bar{x}(0) = x_0,$$

then $\bar{x}(t) = x$. Then we say that (x,t) is in the illuminated region of A (otherwise in the shadow) by analogy with geometrical optics. Clearly $\xi(s) = \bar{x}(t-s)$, $0 \le s \le t$ is a minimizing path for (28) which shows that

$$S(x,t) \equiv 0 \quad \text{in the illuminated region of } A \tag{29}$$

and

$$S(x,t) > 0 \quad \text{in the shadow region of } A \tag{30}$$

This is completely in accord with intuition since being in the "shadow region" means that a large deviation has occurred away from the "illuminated region" where the law of large numbers is valid.

Now, as we said above the expansion (11) fails (i) at $t = 0$ on ∂A and (ii) for $t > 0$ on the boundary of the illuminated region because $\bar{u}_0(x,t)$ (cf. (19)) becomes singular there. In optics this is interpreted as a space-time diffraction effect and a lot of effort has gone into fixing up these singularities [8] and ultimately, obtaining uniformly valid expansions (here this is analogous to, but much simpler than, obtaining uniform expansion for edge diffration [9]). One hopes that these methods will also prove useful in the present context despite the fact that $\bar{S}$ in (11) is real and nonnegative and hence takes care of the major effects under study, and that (8) is a global statement.

What happens[†] if $\bar{F}(x) \equiv 0$ (cf. (5))? Nothing, of course, except that $\bar{x}(t) \equiv x$ in (4) and the rest goes as above. When however $\bar{F}(x) \equiv 0$ it is fruitful to ask what happens to $x^{\varepsilon}(t)$ when $t = O(\varepsilon^{-1})$ i.e. on rescaling $t \to t/\varepsilon$. One then obtains a global approximation again, the diffusion approximation to $x^{\varepsilon}(t)$. It is amusing to note that on the original time scale excursions away from the law of large numbers are large deviations (exponentially small probabilities) while on the new time scale they are of order one effects. It is also interesting and easy to see that the diffusion coefficients in the diffusion approximation are given by

$$a_{ij}(x) = \frac{1}{2} \frac{\partial \omega(k,x)}{\partial k_i \, \partial k_j}\bigg|_{k=0}$$

where of course $\bar{F} \equiv 0$ i.e. $\partial\omega/\partial k|_{k=0} \equiv 0$.

REFERENCES

[1] S.R.S. Varadhan, Asymptotic Probabilities and Differential Equations, Comm. Pure Appl. Math. 19 (1965) pp. 261-286.

[2] A. D. Ventsel and M. I. Freidlin, On small random perturbation of dynamical systems, Russian Math. Surveys 25 (1970) pp. 1-55.

[3] A. D. Ventsel, Rough limit theorems on large deviations for Markov processes I, II, Theor. Prob. Appl. 21 (1976) pp. 227-242, 499-512.

[†] This is quite common with linear equations, after preliminary changes of variables.

[4] M. I. Freidlin, Fluctuation in dynamical systems with averaging, Dokl. Akad. Nauk SSSR (Sov. Math. Dokl.) 17 (1976) 104-108.

[5] M. D. Donsker and S. R. S. Varadhan, Asymptotic evaluation of certain Markov process expectations for large time I, Comm. Pure Appl. Math. 27 (1975) pp. 1-47.

[6] W. Kohler and G. C. Papanicolaou, articles in Proceedings of Conference on Stochastic Differential Equations, J. D. Mason, editor, Academic Press, N. Y., 1977.

[7] N. N. Bogliubov and Yu. Mitropolskii, Asymptotic methods in nonlinear mechanics, Gordon and Breach, N. Y., 1961.

[8] J. B. Keller, Geometrical theory of diffraction, J. Opt. Soc. of Amer. 52 (1962), pp. 116-130.

[9] D. Ahluwalia, J. Boersma and R. M. Lewis, Uniform asymptotic theory of diffraction by a plane screen, SIAM J. Appl. Math. 16 (1968) pp. 783-807.

OPTIMAL FINITE DIMENSIONAL RECURSIVE ESTIMATORS FOR DISCRETE-TIME STOCHASTIC NONLINEAR SYSTEMS

STEVEN I. MARCUS
Department of Electrical Engineering
University of Texas at Austin
Austin, Texas 78712

ABSTRACT

Recursive estimation for nonlinear discrete-time stochastic systems with white Gaussian observation noise is investigated. It is proved that for certain classes of systems, described either by finite Volterra series expansions or by state-linear realizations under certain algebraic conditions, the optimal conditional mean estimator is recursive and finite dimensional. Examples are presented to illustrate the structure of the estimators.

I. INTRODUCTION

This paper is concerned with the optimal recursive estimation of the state of a nonlinear discrete-time stochastic system, given the past observations. Brémaud and Van Schuppen [1]-[3] have recently developed a representation equation for the evaluation of the optimal conditional mean estimator for very general nonlinear discrete-time systems; their results include the cases in which the observations are point processes or nonlinear functions of the state plus Gaussian white noise. However, the optimal estimator is, in general, infinite dimensional. Of course, if the system and observations are linear and the noises are Gaussian, then the Kalman filter provides a recursive finite dimensional estimator. The other known cases of discrete-time finite dimensional estimators involve point process observations [3]-[6]. We also note that finite dimensional estimators for continuous-time linear systems with nonlinear point process observations have been studied in [12] and [13].

In the continuous-time case, we have proved [7]-[9] that the optimal estimators for certain classes of nonlinear stochastic systems with Gaussian observation noise are recursive and finite dimensional. In this paper, we will show that certain classes of discrete-time nonlinear systems with white Gaussian observation noise also admit recursive finite dimensional optimal estimators. The estimators discussed in this paper are similar to those of [7]-[9], but they display some basic differences, including terms involving polynomials in the innovations. These differences are due to the fact that, in opposition to the continuous-time case, not every martingale difference (MD) sequence with respect to the innovations of signals in white Gaussian noise can be written as an MD transform of the innovations [10]. Instead, a more complicated martingale representation theorem, due to Brémaud and Van Schuppen [1]-[3] holds in this case, and gives rise to their nonlinear estimation equations. The derivations in this paper will be based on linear smoothing arguments [11], but the relationship to the results of Brémaud and Van Schuppen will also be discussed in detail.

Due to lack of space, the major results will only be stated here; proofs will appear in a more complete version of the paper. However, an example which illustrates the methodology of proof and the structure of the resulting estimators is presented in Section III.

II. MAIN RESULTS

The general class of systems considered in this paper is given by a Gauss-Markov process x which feeds forward into a nonlinear system with state y. The objective is to estimate x and y given noisy linear observations of x. Specifically, consider the system, for $t \in Z^+$, the set of nonnegative integers,

$$x(t+1) = A(t)x(t) + B(t)w(t) \tag{1}$$

$$y(t+1) = f\big(x(t), y(t), t\big) \tag{2}$$

$$z(t) = C(t)x(t) + v(t) \tag{3}$$

where x(t) is an n-vector, y(t) is an m-vector, z(t) is a p-vector, w(t) and v(t) are zero-mean Gaussian white noise processes with $E[w(t)w'(t)] = I$ and $E[v(t)v'(t)] = R > 0$, x(0) is a Gaussian random variable and $\{x(0), y(0), w(t), v(t)\}$ are independent for all $t \in Z^+$. The specification of the nonlinear system (2), which will determine whether the optimal estimator is finite dimensional, will be given below. Motivation for the study of systems of the form (1)-(3) is considered in [7] and [9].

The optimal estimate, with respect to a wide variety of criteria, of x(t) given the past observations $z^t \triangleq \{z(s),\ s \in Z^+,\ s \leq t\}$, is the conditional mean $\hat{x}(t|t)$ of x(t) given the σ-field F^z_t generated by z^t, also denoted $E[x(t)|z^t]$ or $E^{F^z_t}[x(t)]$. Thus our objective is the recursive computation of $\hat{x}(t|t)$ and $\hat{y}(t|t)$. The computation of $\hat{x}(t|t)$ can be performed by the recursive finite dimensional (linear) Kalman filter. However, the computation of $\hat{y}(t|t)$ requires, in general, an infinite dimensional system of equations. The purpose of this paper is to show that if the nonlinear system (2) is characterized by a certain type of Volterra series expansion or by a certain type of state-affine equation, then $\hat{y}(t|t)$ can be computed with a recursive finite dimensional estimator.

Definition 1: (a) The Volterra series expansion of the i^{th} component of y is given by

$$y_i(t) = w_{0i}(t-1) + \sum_{k=1}^{\infty} \sum_{\ell=0}^{t-1} \sum_{j=1}^{j(k)} w^j_{ki}(t,\ell)\, x^{kj}(\ell) \tag{4}$$

where $\ell = (\ell_1, \dots, \ell_n) \in (Z^+)^n$ (the set of row n-vectors with components in Z^+); $\sum_{\ell=0}^{t-1} \triangleq \sum_{\ell_1, \dots, \ell_n = 0}^{t-1}$; $j(k) \triangleq \binom{n+k-1}{k}$, the binomial coefficient; $x^{k1}(\ell), \dots, x^{k,j(k)}(\ell)$ are the lexicographically ordered monomials $x_1^{k_1}(\ell_1) \dots x_n^{k_n}(\ell_n)$ of degree $k(k = k_1 + \dots + k_n)$ in the components of x; and $w^j_{ki}(t,\ell)$ are the k^{th} order kernels.

(b) A k^{th} order kernel is separable if there exists a positive integer r and functions $\{\gamma_{qi}\}$ such that

$$w^j_{ki}(t,\ell) = \sum_{q=1}^{r} \gamma_{q0}(t)\gamma_{q1}(\ell_1)\dots\gamma_{qn}(\ell_n) \tag{5}$$

(the γ_i will obviously also depend on i and j, and some of the γ_i's may equal 1).

(c) The Volterra series (4) is finite if it contains a finite number of terms--i.e., if there exists $s \varepsilon Z^+$ such that all k^{th} order kernels are zero for $k \geq s$.

Theorem 1: Consider the linear system defined by (1) and (3), and assume that each component of y(t) is described by a finite Volterra series of the form (4), with separable kernels. Then $\hat{y}(t|t)$ can be computed with a finite dimensional recursive system of equations driven by the innovations $\nu(t) \triangleq z(t) - C(t)A(t-1)\,\hat{x}(t-1|t-1)$.

It can be shown, using methods similar to [14]-[16], that every shift-invariant finite Volterra series of the form (4) satisfying realizability (or separability) conditions has a state-affine realization [15].

$$\tilde{y}(t+1) = F\tilde{y}(t) + \left[\sum_{k=1}^{\delta}\sum_{j=1}^{j(k)} N_{kj}x^{kj}(t)\right]\tilde{y}(t) + \sum_{k=1}^{\sigma}\sum_{j=1}^{j(k)} b_{kj}x^{kj}(t) \qquad (6)$$

$$y(t) = H\tilde{y}(t) \qquad (7)$$

where δ and σ are positive integers, $x^{kj}(t) = x_1^{k_1}(t) \ldots x_n^{k_n}(t)$ with $k = k_1 + \ldots + k_n$, and F and N_{kj} are partitioned block matrices of the form

$$F = \begin{bmatrix} F_{11} & & & & & \\ F_{21} & \cdot & & & 0 & \\ \cdot & & \cdot & & & \\ \cdot & & & \cdot & & \\ \cdot & & & & \cdot & \\ F_{d1} & \cdot & \cdot & \cdot & \cdot & F_{dd} \end{bmatrix} \qquad N_{kj} = \begin{bmatrix} 0 & & & & \\ N^{21} & \cdot & & 0 & \\ \cdot & & \cdot & & \\ \cdot & & & \cdot & \\ \cdot & & & & \cdot \\ N^{d1} & \cdot & \cdot & N^{d,d-1} & 0 \end{bmatrix} \qquad (8)$$

Equation (6) can obviously be put into state-linear form (all $b_{kj} = 0$) by augmenting the state with the equation

$$u(t+1) = u(t) \; ; \quad u(0) = 1 \qquad (9)$$

For such systems, the following theorem can be proved. First we define $\{F,N_{kj}\}_{AA}$ to be the associative algebra generated by F and N_{kj}, $k=1,\ldots,\delta$; $j=1,\ldots,j(k)$ [14],[18]; that is, $\{F,N_{kj}\}_{AA}$ is the smallest vector space of matrices which is closed under multiplication and contains F and $\{N_{kj}\}$. An associative algebra A is said to be nilpotent if there exists a positive integer s such that the product of any s elements of A vanishes [14],[18].

Theorem 2: Consider the linear system defined by (1) and (3), and assume that y(t) is the output of a state-linear system of the form (6)-(7) (i.e., all $b_{kj} = 0$). Assume that the ideal in $\{F,N_{kj}\}_{AA}$ generated by the matrices $\{N_{kj}\}$ is nilpotent. Then $\hat{y}(t|t)$ can be computed with a finite dimensional recursive estimator.

The basic technique employed in the proofs of Theorems 1 and 2 is, as in [7]-[9], the augmentation of the state of the original system with additional states which arise in a natural way (as demonstrated in the example of the next section, they are smoothed statistics of the state). For the classes of systems considered here, it is shown that only a finite number of additional states (smoothed statistics) are required.

III. AN EXAMPLE

In this section we discuss an example which illustrates the basic concepts and results stated above; we show that the optimal estimator is in fact recursive and finite dimensional. Consider the system (of the form (1)-(3)) described by

$$\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \end{bmatrix} = \begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix} \triangleq Ax(t) + w(t) \tag{10}$$

$$y(t+1) = \gamma\, y(t) + x_1(t)\, x_2(t) \tag{11}$$

$$\begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} \tag{12}$$

where $\alpha, \beta \neq 0$; $|\gamma| < 1$; w_1, w_2, v_1, and v_2 are independent, zero mean, unit variance, Gaussian white noise processes; $y(0) = 0$; and $x_1(0)$ and $x_2(0)$ are independent Gaussian random variables.

The conditional mean estimate $\hat{x}(t|t)$ and nonrandom (precomputable) error covariance matrix $P(t|t)$ are of course computed by the (finite dimensional) Kalman filter [11, Eqns. (5.48)-(5.51)] for (10),(12). Since $x_1(0)$ and $x_2(0)$ are independent, it is easy to show that $P_{12}(t|t) = P_{21}(t|t) = 0$ for all $t \in Z^+$. Thus we have that

$$\hat{x}(t+1|t+1) = A\hat{x}(t|t) + K(t+1)\, \tilde{z}(t+1) \tag{13}$$

where

$$\tilde{z}(t+1) = z(t+1) - A\hat{x}(t|t) \tag{14}$$

is the innovations process, $K(t+1) = P(t+1|t+1)$, and

$$P(t+1|t) \triangleq \begin{bmatrix} P_{11}(t+1|t) & 0 \\ 0 & P_{22}(t+1|t) \end{bmatrix} = \begin{bmatrix} \alpha^2 P_{11}(t|t)+1 & 0 \\ 0 & \beta^2 P_{22}(t|t)+1 \end{bmatrix} \tag{15}$$

$$P(t+1|t+1) \triangleq \begin{bmatrix} P_{11}(t+1|t+1) & 0 \\ 0 & P_{22}(t+1|t+1) \end{bmatrix} = \begin{bmatrix} \dfrac{\alpha^2 P_{11}(t|t)+1}{\alpha^2 P_{11}(t|t)+2} & 0 \\ 0 & \dfrac{\beta^2 P_{22}(t|t)+1}{\beta^2 P_{22}(t|t)+2} \end{bmatrix} \tag{16}$$

The estimate $\hat{y}(t|t)$ provides the interesting results. Now,

$$\hat{y}(t+1|t+1) = \gamma\hat{y}(t|t+1) + E^{t+1}\left[x_1(t)\, x_2(t)\right]$$

$$= \gamma E^{t+1}\left[\sum_{i=0}^{t-1} \gamma^{t-1-i}\, x_1(i)x_2(i)\right] + \hat{x}_1(t|t+1)\hat{x}_2(t|t+1) \tag{17}$$

where, e.g., $\hat{x}(i|j) \triangleq E\left[x(i)|z^j\right] \triangleq E^j\left[x(i)\right]$.

In order to proceed further, certain smoothing results are needed.

Lemma 1 [11, Eqs. (6.76)-(6.78)]: The optimal fixed-point smoothed estimate $\hat{x}(i|t)$ is given by

$$\begin{aligned}\hat{x}(i|t+1) &= \hat{x}(i|t) + M(i,t+1)\left[\hat{x}(t+1|t+1) - \hat{x}(t+1|t)\right] \\ &= \hat{x}(i|t) + M(i,t+1)\ K(t+1)\ \tilde{z}(t+1) \qquad (18)\end{aligned}$$

for $t = i,\ i+1,\ i+2,\ \ldots$, and initial condition $\hat{x}(i|i)$. The smoothing gain matrix is given by

$$M(i,t+1) = \prod_{j=i}^{t} F(j) \triangleq F(i) \ldots F(t) \tag{19}$$

where

$$F(j) = P(j|j)\ A'\ P^{-1}(j+1|j) \tag{20}$$

In this example,

$$F(j) = \begin{bmatrix} \alpha P_{11}(j|j)\ P_{11}^{-1}(j+1|j) & 0 \\ 0 & \beta P_{22}(j|j)\ P_{22}^{-1}(j+1|j) \end{bmatrix} \tag{21}$$

and $M(i,t+1) \triangleq B(i,t+1)\ K(t+1)$ is a diagonal matrix (nonrandom and precomputable) with

$$\begin{aligned} M_{11}(i,t+1) &= \frac{\alpha^{t-i+1}\ P_{11}(i|i) \ldots P_{11}(t|t)}{P_{11}(i+1|i) \ldots P_{11}(t|t-1)\left[P_{11}(t+1|t) + 1\right]} \\ M_{22}(i,t+1) &= \frac{\beta^{t-i+1}\ P_{22}(i|i) \ldots P_{22}(t|t)}{P_{22}(i+1|i) \ldots P_{22}(t|t-1)\left[P_{22}(t+1|t) + 1\right]} \end{aligned} \tag{22}$$

From (18) and (22), the second term in (17) can be expressed as

$$\begin{aligned} &\left[\hat{x}_1(t|t) + \frac{\alpha P_{11}(t|t)}{\alpha^2 P_{11}(t|t) + 2}\ \tilde{z}_1(t+1)\right]\left[\hat{x}_2(t|t) + \frac{\beta P_{22}(t|t)}{\beta^2 P_{22}(t|t) + 2}\ \tilde{z}_2(t+1)\right] \\ &= \left[\hat{x}_1(t|t) + M_{11}(t,t+1)\ \tilde{z}_1(t+1)\right]\left[\hat{x}_2(t|t) + M_{22}(t,t+1)\ \tilde{z}_2(t+1)\right] \qquad (23)\end{aligned}$$

Since $P(t|t)$ is nonrandom, (23) is finite dimensionally computable (FDC), as a product of terms from the Kalman filter (13). Now consider the first term in (17):

$$E^{t+1}\left[\sum_{i=0}^{t-1} \gamma^{t-1-i} \; x_1(i)\, x_2(i)\right] = \sum_{i=0}^{t-1} \gamma^{t-1-i} \; \hat{x}_1(i|t+1)\, \hat{x}_2(i|t+1) \tag{24}$$

$$= \sum_{i=0}^{t-1} \gamma^{t-1-i} \left[\hat{x}_1(i|t) + M_{11}(i,t+1)\, \tilde{z}_1(t+1)\right] \left[\hat{x}_2(i|t) + M_{22}(i,t+1)\, \tilde{z}_2(t+1)\right] \tag{25}$$

$$= \sum_{i=0}^{t-1} \gamma^{t-1-i} \Big[\hat{x}_1(i|t)\, \hat{x}_2(i|t) + M_{11}(i,t+1)\, \hat{x}_2(i|t)\, \tilde{z}_1(t+1)$$

$$+ M_{22}(i,t+1)\, \hat{x}_1(i|t)\, \tilde{z}_2(t+1) + M_{11}(i,t+1)\, M_{22}(i,t+1)\, \tilde{z}_1(t+1)\, \tilde{z}_2(t+1)\Big] \tag{26}$$

Here, (24) follows from the form of (10) and the independence of $x_1(0)$ and $x_2(0)$; and (25) follows from (18). The last term in (26) is

$$\left[\sum_{i=0}^{t-1} \gamma^{t-1-i} \; M_{11}(i,t+1)\, M_{22}(i,t+1)\right] \tilde{z}_1(t+1)\, \tilde{z}_2(t+1) \tag{27}$$

which is FDC since M(i,t+1) is nonrandom. The first term in (26) is equal to

$$E^t\left[\sum_{i=0}^{t-1} \gamma^{t-1-i} \; x_1(i)\, x_2(i)\right] = \hat{y}(t|t)\ . \tag{28}$$

The second and third terms in (26) are equal to

$$\tilde{z}_1(t+1) \cdot E^t\left[\sum_{i=0}^{t-1} \gamma^{t-1-i} \; M_{11}(i,t+1)\, x_2(i)\right]$$

$$+ \tilde{z}_2(t+1) \cdot E^t\left[\sum_{i=0}^{t-1} \gamma^{t-1-i} \; M_{22}(i,t+1)\, x_1(i)\right] \triangleq \tilde{z}'(t+1)\; E^t\begin{bmatrix} M_{11}(t,t+1)\ \xi_2(t) \\ M_{22}(t,t+1)\ \xi_1(t) \end{bmatrix} \tag{29}$$

where

$$\begin{bmatrix} \xi_1(t+1) \\ \xi_2(t+1) \end{bmatrix} = \begin{bmatrix} \gamma F_{22}(t) & 0 \\ 0 & \gamma F_{11}(t) \end{bmatrix} \begin{bmatrix} \xi_1(t) \\ \xi_2(t) \end{bmatrix} + \begin{bmatrix} F_{22}(t) & 0 \\ 0 & F_{11}(t) \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \tag{30}$$

$$\xi_1(0) = \xi_2(0) = 0$$

and (see (21))

$$F_{11}(t) = \alpha P_{11}(t|t)/P_{11}(t+1|t)\ ; \quad F_{22}(t) = \beta P_{22}(t|t)/P_{22}(t+1|t) \tag{31}$$

In other words, the argument of the conditional expectation in (29) can be realized as the output of a finite dimensional linear system with state $\xi(t) = \left[\xi_1(t),\ \xi_2(t)\right]'$ satisfying (30).

Thus the finite dimensional (5-state) recursive estimator for the system (10)-(12) is constructed as follows. First we augment the state x of (10) with the state ξ of (30). Then the Kalman filter for the linear system (10),(30), with observations (12), computes the conditional expectations $\hat{x}(t|t)$ and $\hat{\xi}(t|t)$. Finally, from (17),(23),(26)-(29), the recursive computation of $\hat{y}(t|t)$ is accomplished by the scalar equation

$$\begin{aligned}
\hat{y}(t+1|t+1) &= \gamma\hat{y}(t|t) + \hat{x}_1(t|t)\,\hat{x}_2(t|t) \\
&\quad + M_{11}(t,t+1)\left[\hat{x}_2(t|t) + \gamma\hat{\xi}_2(t|t)\right]\tilde{z}_1(t+1) \\
&\quad + M_{22}(t,t+1)\left[\hat{x}_1(t|t) + \gamma\hat{\xi}_1(t|t)\right]\tilde{z}_2(t+1) \\
&\quad + \left[\sum_{i=0}^{t}\gamma^{t-i}\,M_{11}(i,t+1)\,M_{22}(i,t+1)\right]\tilde{z}_1(t+1)\,\tilde{z}_2(t+1) \\
\hat{y}(0|0) &= 0
\end{aligned} \tag{32}$$

IV. DISCUSSION

The finite dimensional nonlinear estimator of the previous section bears a resemblance to similar continuous time estimators [7]-[9], but it also has some distinguishing features. Notice that the last term in (32) involves a product of the innovations -- a phenomenon which never occurs in continuous-time systems involving "Gaussian white noise." It can be shown that if equation (11) for y involves higher order polynomials in x, then the estimator will contain higher order polynomials in the innovations.

The reason that such polynomials arise in this case (as opposed to the discrete-time point process observation [4]-[6] or continuous-time [7]-[9] cases) is that not every non-Gaussian martingale difference (MD) sequence with respect to the innovations of signals in white Gaussian noise can be written as an MD transform of the innovations process [10]. However, Brémaud and Van Schuppen [2, Theorem 3.4] prove a more general martingale representation theorem, which states that, if M(t) is any real-valued martingale adapted to the σ-fields $F_t^z \triangleq \sigma\left(z(s);\ s \in Z^+,\ s \leq t\right)$ generated by a given $\mathbb{R}^p$-valued process z, then there exists an F_t^z-predictable $\mathbb{R}^p$-indexed process K such that, for all $t \in Z^+$, $E\left[\left|K\left(t,z(t)\right)\right|\right] < \infty$ and

$$M(t) = E\left[M(0)\right] + \sum_{j=0}^{t}\int_{\mathbb{R}^p} K(j,z)\,(\bar{z}-\mu^z)(j,dz) \tag{33}$$

$$\triangleq E\left[M(0)\right] + \sum_{j=0}^{t}\left[K\left(j,z(j)\right) - E^{F_{j-1}^z}\left[K\left(j,z(j)\right)\right]\right] \tag{34}$$

where the innovations kernel $(\bar{z}-\mu^z)$ is defined by (34). As we will demonstrate, this formula can give rise, if z is Gaussian, to polynomials in the innovations $\tilde{z}(t+1) \triangleq z(t+1) - E^{F_t^z}\left[z(t+1)\right]$. Furthermore, this martingale representation theorem is then applied to the estimation problem in [3, Theorem 3.4]. For the system (10)-(12), the estimator representation of [3, Theorem 3.4] implies the existence of a F_t^z-predictable $\mathbb{R}^2$-indexed process K such that

$$\hat{y}(t+1|t+1) = \gamma\hat{y}(t|t) + \hat{x}_1(t|t)\,\hat{x}_2(t|t) + \left\{K\big(t+1, z(t+1)\big) - E^{F_t^z}\left[K\big(t+1, z(t+1)\big)\right]\right\} \tag{35}$$

where K satisfies the defining equations [3, eqs. (3.4)-(3.5)]. This result is due to the application of (34) to the martingale $\overline{M}(t)$ defined by

$$\begin{aligned}\Delta\,\overline{M}(t+1) \triangleq \overline{M}(t+1) - \overline{M}(t) &= \hat{y}(t+1|t+1) - E^{F_t^z}\left[y(t+1)\right]\\ &= \hat{y}(t+1|t+1) - \gamma\hat{y}(t|t) - \hat{x}_1(t|t)\,\hat{x}_2(t|t)\end{aligned} \tag{36}$$

It can be shown that $K\big(t+1, z(t+1)\big)$ in (35) is given by

$$\begin{aligned}K\big(t+1, z(t+1)\big) &= M_{11}(t,t+1)\left[\hat{x}_2(t|t) + \gamma\hat{\xi}_2(t|t)\right] z_1(t+1)\\ &+ M_{22}(t,t+1)\left[\hat{x}_1(t|t) + \gamma\hat{\xi}_1(t|t)\right] z_2(t+1)\\ &+ \left[\sum_{i=0}^{t} \gamma^{t-i}\, M_{11}(i,t+1)\, M_{22}(i,t+1)\right]\\ &\cdot \left[z_1(t+1)\, z_2(t+1) - z_1(t+1)\,\beta\hat{x}_2(t|t) - z_2(t+1)\,\alpha\hat{x}_1(t|t)\right]\end{aligned} \tag{37}$$

which contains a quadratic term in z(t+1). Performing the indicated operations in (35) gives precisely the equation (32), with quadratic terms in the innovations.

Thus the results of Brémaud and Van Schuppen provide further insight into the estimation problems considered here -- the appearance of polynomials in the innovations can be attributed to the more complicated martingale representation (34). The same reasoning implies that higher order polynomials in x(t) in (2) give rise to higher order polynomials in the innovations in the recursive estimation equations. However, direct application of [3, Theorem 3.4] for the problems considered here is quite difficult; the application of smoothing formulas, as in Section III, provides a simpler method of obtaining the estimation equations.

V. STEADY-STATE BEHAVIOR

We now consider the steady-state performance of the optimal estimator of Section III. Since the linear system (10),(12) is controllable and observable, the Riccati equation (15)-(16) has a unique positive-definite steady state solution [19]; a simple computation shows that

$$\lim_{t\to\infty} P(t|t) \triangleq \overline{P} = \begin{bmatrix} \dfrac{\alpha^2 - 2 + \sqrt{\alpha^4 + 4}}{2\,\alpha^2} & 0 \\ 0 & \dfrac{\beta^2 - 2 + \sqrt{\beta^4 + 4}}{2\,\beta^2} \end{bmatrix} \tag{38}$$

$$\lim_{t\to\infty} P(t+1|t) \triangleq P = \begin{bmatrix} \frac{1}{2}\left(\alpha^2 + \sqrt{\alpha^4 + 4}\right) & 0 \\ 0 & \frac{1}{2}\left(\beta^2 + \sqrt{\beta^4 + 4}\right) \end{bmatrix} \tag{39}$$

Thus, in steady-state, the augmented linear system (10),(12),(30) is time-invariant. Now consider the eigenvalues of (30) in steady-state (see (31)):

$$\begin{aligned} \gamma F_{11} &= \gamma\alpha\bar{P}_{11}/P_{11} = \gamma\alpha/(1+P_{11}) \\ \gamma F_{22} &= \gamma\beta\bar{P}_{22}/P_{22} = \gamma\beta/(1+P_{22}) \end{aligned} \tag{40}$$

It is easily shown that $|\alpha/(1+P_{11})| < 1$ and $|\beta/(1+P_{22})| < 1$; since $|\gamma| < 1$ by assumption, these eigenvalues both have magnitudes less than one. Hence the augmented system linear system (10),(12),(30) is stabilizable and detectable. Let the Kalman filter error covariance matrix of the augmented state $[x(t),\xi(t)]$ be denoted by $S(t+1|t)$. Then [19] the Riccati equation satisfied by $S(t+1|t)$ has a unique positive definite steady-state solution S (notice that $S_{11} = P_{11}$ and $S_{22} = P_{22}$).

The steady-state Kalman filter for the augmented system (10),(12),(30) is then

$$\begin{bmatrix} \hat{x}_1(t+1|t+1) \\ \hat{x}_2(t+1|t+1) \\ \hat{\xi}_1(t+1|t+1) \\ \hat{\xi}_2(t+1|t+1) \end{bmatrix} = \begin{bmatrix} \alpha & 0 & 0 & 0 \\ 0 & \beta & 0 & 0 \\ \frac{\beta}{1+P_{22}} & 0 & \frac{\beta\gamma}{1+P_{22}} & 0 \\ 0 & \frac{\alpha}{1+P_{11}} & 0 & \frac{\alpha\gamma}{1+P_{11}} \end{bmatrix} \begin{bmatrix} \hat{x}_1(t|t) \\ \hat{x}_2(t|t) \\ \hat{\xi}_1(t|t) \\ \hat{\xi}_2(t|t) \end{bmatrix} + \begin{bmatrix} P_{11} & 0 \\ 0 & P_{22} \\ S_{13} & 0 \\ 0 & S_{24} \end{bmatrix} \tilde{z}(t+1) \tag{41}$$

where

$$S_{13} = \frac{\alpha\beta P_{11}}{(P_{11}+1)(P_{22}+1) - \alpha\beta\gamma} \;; \quad S_{24} = \frac{\alpha\beta P_{22}}{(P_{11}+1)(P_{22}+1) - \alpha\beta\gamma}$$

(here P_{11} and P_{22} are defined in (39)). The description of the optimal steady estimator is completed with the equation for $\hat{y}(t|t)$ (see (32)):

$$\begin{aligned} \hat{y}(t+1|t+1) &= \gamma\hat{y}(t|t) + \hat{x}_1(t|t)\,\hat{x}_2(t|t) \\ &+ \frac{\beta P_{11}}{(P_{11}+1)(P_{22}+1)}\left[\hat{x}_2(t|t) + \gamma\hat{\xi}_2(t|t)\right]\tilde{z}_1(t+1) \\ &+ \frac{\alpha P_{22}}{(P_{11}+1)(P_{22}+1)}\left[\hat{x}_1(t|t) + \gamma\hat{\xi}_1(t|t)\right]\tilde{z}_2(t+1) \\ &+ \frac{\alpha\beta P_{11}P_{22}}{(P_{11}+1)(P_{22}+1)[(P_{11}+1)(P_{22}+1) - \alpha\beta\gamma]}\,\tilde{z}_1(t+1)\,\tilde{z}_2(t+1) \\ \hat{y}(0|0) &= 0 \end{aligned} \tag{42}$$

which is a nonlinear, time-invariant equation.

ACKNOWLEDGEMENT

This research was supported in part by the Joint Services Electronics Program under Contract F49620-77-C-0101 and in part by the National Science Foundation under Grant ENG 76-11106.

References

1. J.H. Van Schuppen, "An introduction to discrete time martingales and filtering problems," *Proc. 14th Allerton Conf. Circuits and Systems,* 1976, pp. 1-9.
2. P.M. Brémaud and J.H. Van Schuppen, "Discrete time processes: Preliminaries and Part I: Martingale calculus and innovations kernels," preprint, June 1976.
3. ——— "Discrete time processes II: Estimation," preprint, June 1976.
4. P.M. Brémaud, "Countable state space signals and multivariate point process observations: Application to ALOHA," preprint, June 1976.
5. A. Segall, "Recursive estimation from discrete-time point processes," *IEEE Trans. Inform. Theory,* vol. IT-22, pp. 422-431, July 1976.
6. M.V. Vaca and S.A. Tretter, "Optimal estimation for discrete-time jump process," *Proc. 1977 Johns Hopkins Conf. on Inform. Sciences and Systems,* p. 268.
7. S.I. Marcus, "Estimation and analysis of nonlinear stochastic systems," Ph.D. Thesis, Dept. of Electrical Engineering, M.I.T., Cambridge, MA., June 1975; also M.I.T. Electronic Systems Lab. Report No. ESL-R-601, June 1975.
8. S.I. Marcus and A.S. Willsky, "Algebraic structure and finite dimensional nonlinear estimation," in *Mathematical Systems Theory,* G. Marchesini and S.K. Mitter, eds. New York: Springer-Verlag, 1976, pp. 301-311.
9. S.I. Marcus and A.S. Willsky, "Algebraic structure and finite dimensional nonlinear estimation," *SIAM J. Math. Anal.,* to appear.
10. A. Segall, "Stochastic processes in estimation theory," *IEEE Trans. Inform. Theory,* vol. IT-22, pp. 275-286, May 1976.
11. J.S. Meditch, *Stochastic Optimal Linear Estimation and Control.* New York: McGraw-Hill, 1969.
12. D.L. Snyder and P.M. Fishman, "How to track a swarm of fireflies by observing their flashes," *IEEE Trans. Inform. Theory,* vol. IT-21, pp. 692-695, November 1975.
13. I.B. Rhodes and D.L. Snyder, "Estimation and control performance for space-time point-process observations," *IEEE Trans. Automatic Control,* vol. AC-22, pp. 338-345, June 1977.
14. R.W. Brockett, "Volterra series and geometric control theory," *Automatica,* vol. 12, pp. 167-176, 1976.
15. E.G. Gilbert, "Bilinear and 2-power input-output maps: Finite dimensional realizations and the role of functional series," *Proc. 1977 Joint Automatic Control Conf.,* pp. 917-924.
16. E.D. Sontag, "On the internal realization of polynomial response maps," Ph.D. Dissertation, University of Florida, Gainesville, 1976.
17. T.J. Tarn, J. Nonoyama, and L.S. Oei, "Algebraic structure of discrete-time polynomial systems," *Proc. 1976 IEEE Conf. on Decision and Control,* pp. 826-835.
18. A. Abian, *Linear Associative Algebras.* New York: Pergamon, 1971.
19. G.A. Hewer, "Analysis of a discrete matrix Riccati equation of linear control and Kalman filtering," *J. Math. Anal. Appl.,* vol. 42, pp. 226-236, 1973.

QUANTUM DETECTION AND ESTIMATION THEORY(1)

SANJOY K. MITTER(2) and S.K. YOUNG(3)

(2) Department of Electrical Engineering and Computer Science
MIT, Cambridge, MA 02139

(3) Department of Mathematics, MIT, Cambridge, MA 02139
Now at BDM Corporation, McLean, VA

1. INTRODUCTION

In classical communication theory, the message to be transmitted is modulated and the resulting signal propagates through a given channel to produce a received waveform. The function of the receiver is to recover the signal from the received waveform, perhaps in an optimum manner, optimum being defined by some performance criterion. The input to the receiver may have some additive noise added to the received waveform. It is assumed that the receiver can be constructed independently of the model of the received waveform and the additive noise. Moreover, it is assumed that the optimum receiver can be physically realized.

In communication at optical frequencies neither of these two assumptions are valid. No matter what measurement we make of the received field, the outcome is random whose statistics depend on the measurement being made. This is a reflection of the laws of quantum physics. Furthermore, there is no guarantee that the measurement characterizing the receiver can be actually implemented.

In this paper, we announce some results on the M-any quantum detection problem. Full details will be published elsewhere [1]. For related work, see [2] and [3]. The proof in [2] appears to be incorrect and in [3], a complete duality theory is not presented.

It will be assumed that the reader is familiar with the notions of convex analysis in infinite dimensional spaces as for example, presented in [4].

In the classical formulation of detection theory (Bayesian hypothesis testing) it is desired to decide which of n possible hypotheses $H_1, \ldots, H_n$ is true, based on observation of a random variable whose probability distribution depends on the several hypotheses. The decision entails certain costs that depend on which hypothesis is selected and which hypothesis corresponds to the true state of the system. A decision procedure or strategy prescribes which hypothesis is to be chosen for each possible outcome of the observed data; in general, it may be necessary to use a randomized strategy which specifies the probabilities with which each hypothesis should be chosen as a function of the observed data. The detection problem is to determine an optimal decision strategy.

In the quantum formulation of the detection problem, each hypothesis H_j corresponds to a possible state ρ_j of the quantum system under consideration. Unlike the classical situation, however, it is not possible to measure all relevant variables associated with the state of the system and to specify meaningful probability distributions for the resulting values. For the quantum detection problem it is necessary to specify not only the procedure for processing the experimental data, but also what data to measure in the first place. Hence the quantum detection problem involves determining the entire measurement process, or, in mathematical terms, determining the probability operator measure corresponding to the measurement process (for a discussion see [5]).

---

(1) This research has been supported by NSF Grant #ENG76-02860.

2. FORMULATION OF THE QUANTUM DETECTION PROBLEM

Let H be a separable complex Hilbert space corresponding to the physical variables of the system under consideration. There are n hypotheses $H_1,\dots,H_n$ about the state of the system, each corresponding to a different density operator ρ_j; every ρ_j is a nonnegative definite selfadjoint trace-class operator on H with trace 1 and is the analog of the distribution functions in the classical problem. Let S denote the set $S = \{1,\dots,n\}$. A general decision strategy is determined by a probability operator measure (POM) $m : 2^S \to \mathcal{L}_s(H)_+$ [(1)]; in this case the POM effecting the decision needs only n components $m_1,\dots,m_n$ where each m_j is a positive selfadjoint bounded linear operator on H and

$$\sum_{i=1}^{n} m_i = I \quad . \tag{1}$$

The measurement outcome is an integer $i \in S$; the conditional probability that the hypothesis H_i is chosen when the state of the system is ρ_j is given by

$$\Pr\{i|j\} = \operatorname{tr}(\rho_j m_i) \qquad i,j = 1,\dots,m \quad . \tag{2}$$

We remark that it is crucial here to formulate the problem in terms of general probability operator measures rather than resolutions of the identity. For example, an instrument which simply chooses an arbitrary hypothesis with probability $1/n$ without even interacting with the system corresponds to a measurement process with the POM given by

$$m_j = I/n \quad ;$$

these are certainly not projections.

We denote by C_{ij} the cost associated with choosing hypothesis H_i when H_j is true. For a specified decision procedure effected by the POM $\{m_1,\dots,m_n\}$, the risk function is the conditional expected cost given that the system is in the state ρ_j , i.e.,

---

(1) $\mathcal{L}_s(H)_+$ denotes the space of bounded positive selfadjoint operators on H.

$$R_m(j) = \mathrm{tr}[\rho_j \sum_{i=1}^{n} C_{ij} m_i] \ .$$

If now μ_j specifies a prior probability for hypothesis H_j , the Bayes cost is the posterior expected cost

$$R_m = \sum_{j=1}^{n} R_m(j)\mu_j = \mathrm{tr} \sum_{i=1}^{n} f_i m_i \tag{3}$$

where f_i is the selfadjoint trace-class operator

$$f_i = \sum_{j=1}^{n} c_{ij}\mu_j\rho_j \qquad i = 1,\dots,n \quad . \tag{4}$$

The quantum detection problem is to find $m_1,\dots,m_n$ so as to minimize (3) subject to the constraint (1) and subject to the condition that the operators m_j be selfadjoint and nonnegative definite, $m_j \geq 0$.

3. THE QUANTUM DETECTION PROBLEM AND ITS DUAL

Let H be a complex Hilbert space. The real linear space of compact self-adjoint operators $K_s(H)$ with the operator norm is a Banach space whose dual is isometrically isomorphic to the real Banach space $\tau_s(H)$ of self-adjoint trace-class operators with the trace norm, i.e., $K_s(H)^* = \tau_s(H)$ under the duality

$$\langle A,B\rangle = \mathrm{tr}(AB) \leq |A|_{tr}|B| \qquad A\varepsilon\tau_s(H),\ B\varepsilon K_s(H) \ .$$

Here, $|B| = \sup\{|B\phi| : \phi\varepsilon H, |\phi| \leq 1\} = \sup\{\mathrm{tr}(AB) : A\varepsilon\tau_s(H), |A|_{tr} \leq 1\}$ and $|A|_{tr}$ is the trace norm $\sum_i |\lambda| < +\infty$ where $A\varepsilon\tau_s(H)$ and $\{\lambda_i\}$ are the eigenvalues of A repeated according to multiplicity. The dual of $\tau_s(H)$ with the trace norm is isometrically isomorphic to the space of all linear bounded self-adjoint operators, i.e. $\tau_s(H)^* = L_s(H)$ under the duality

$$\langle A,B\rangle = \mathrm{tr}(AB) \qquad A\varepsilon\tau_s(H),\ B\varepsilon L_s(H).$$

Moreover the orderings are compatible in the following sense. If $K_s(H)_+$, $\tau_s(H)_+$, and $L_s(H)_+$ denote the closed convex cones of nonnegative definite operators in $K_s(H)$, $\tau_s(H)$, and $L_s(H)$ respectively, then

$$[K_s(H)_+]^+ = \tau_s(H)_+ \quad \text{and} \quad [\tau_s(H)_+]^+ = L_s(H)_+$$

where the associated dual spaces are to be understood in the sense defined above.

Let f_j be given elements of $\tau_s(H)$ (as defined in (4)), $j=1,\ldots n$. Define the functionals $F_j\colon L_s(H) \to \overline{R}$ by

$$F_j(A) = \delta_{\geq o}(A) + \operatorname{tr}(f_j A) \qquad A \varepsilon L_s(H),\ j = 1,\ldots,n. \tag{5}$$

where $\delta_{\geq o}(\cdot)$ denotes the indicator function for the set $L_s(H)_+$ of non-negative definite operators, i.e. $\delta_{\geq o}(A)$ is 0 if $A \geq 0$ and $+\infty$ otherwise. Each F_j is proper convex and w*-lower semicontinuous on $L_s(H)$, since $L_s(H)_+$ is a w*-closed convex cone and $A \mapsto \operatorname{tr}(f_j A)$ is a continuous (in fact w*-continuous) linear functional on $L_s(H)$. Define the function $G\colon L_s(H) \to \overline{R}$ by

$$G(A) = \delta_{\{o\}}(A), \quad A \varepsilon L_s(H), \tag{6}$$

that is $G(A)$ is 0 if $A = 0$ and $G(A)$ is $+\infty$ if $A \neq 0$; G is trivially convex and lower semicontinuous. Let $m = (m_1,\ldots,m_n)$ denote an element of $L_s(H)^n$, the Cartesian product of n copies of $L_s(H)$. Then the quantum detection problem (3) may be written

$$P = \inf\left\{ \sum_{j=1}^{n} F_j(m_j) + G(I-Lm) : m = (m_1,\ldots,m_n) \varepsilon L_s(H)^n \right\} \tag{7}$$

where $L\colon L_s(H)^n \to L_s(H)$ is the continuous linear operator

$$L(m) = \sum_{j=1}^{n} m_j, \quad m \varepsilon L_s(H)^n. \tag{8}$$

We consider a family of perturbed problems defined by

$$P(A) = \inf\left\{ \sum_{j=1}^{n} F_j(m_j) + G(A-Lm) : m \varepsilon L_s(H)^n \right\}, \quad A \varepsilon L_s(H). \tag{9}$$

$P(\cdot)$ is a convex function $L_s(H) \to \overline{R}$ and $P = P(I)$. Note that we are taking perturbations in the equality constraint, i.e. the problem $P(A)$ requires that every feasible m satisfy $Lm = A$. We remark that $G(\cdot)$ is nowhere continuous, so that there is certainly no Kuhn-Tucker point $\overline{m}$ such that $G(\cdot)$ is continuous at $L\overline{m}$.

In order to construct the dual problem corresponding to the family of perturbed problems (9) we must calculate the conjugate functions of F_j and G. We would like to pose the dual problem in the space $\tau_s(H)$, so we consider $L_s(H) = \tau_s(H)^*$ and compute the pre-conjugates of F_j, G. It can be shown that the dual problem is

$$({}^*P)(I) = \sup\{\operatorname{tr}(y) : y \varepsilon \tau_s(H),\ f_j - y \geq 0 \quad j = 1,\ldots,n\}. \tag{10}$$

It is not too difficult to show that the primal problem has a solution and that there is no duality gap. The difficult part is to show that the dual problem has solutions. It turns out that the level sets of the dual cost functions are bounded in $\tau_s(H)$ but not weakly compact. This suggests that we imbed $\tau_s(H)$ in its bidual $\tau_s(H)^{**} = \mathcal{L}_s(H)^*$ and extend the dual problem to the larger space; it will then turn out that there are solutions in $\tau_s(H)$. This approach works because $\tau_s(H)$ has a natural topological complement as a subset of $\mathcal{L}_s(H)^*$.

Proposition 1. $\mathcal{L}_s(H)^* = \tau_s(H) \oplus_1 (J\mathcal{K}_s(H))$ where J is the canonical imbedding of $\mathcal{K}_s(H)$ in $\mathcal{L}_s(H)$. In other words, every bounded linear functional y on $\mathcal{L}_s(H)$ may be uniquely represented in the form $y = y_{ac} \oplus y_{sg}$ where $y_{ac} \in \tau_s(H)$ and $y_{sg} \in \mathcal{K}_s(H)$, and

$$y(A) = \mathrm{tr}(y_{ac}A) + y_{sg}(A),\ A \in \mathcal{L}_s(H)$$

$$|y| = |y_{ac}|_{tr} + |y_{sg}|. \blacksquare$$

Before calculating the dual problem, it is necessary to determine what the positive linear functions look like in terms of the decomposition provided by Proposition 1.

Proposition 2. Let $y \in \mathcal{L}_s(H)^*$. Then $y \in [\mathcal{L}_s(H)_+]^+$ iff $y_{ac} \in \tau_s(H)_+$ and $y_{sg} \in [\mathcal{L}_s(H)_+]^+$.$\blacksquare$

It can be shown that in this enlarged space the dual problem $*(P^*)(I) = \sup\limits_y [y(I)-P^*(y)]$ is given by

$$*(P^*)(I) = \sup\{\mathrm{tr}(y_{ac})+y_{sg}(I) : y \in \mathcal{L}_s(H)^*,\ y_{sg} \le 0,\ y_{ac} \le f_j \quad j = 1,\dots,n\}.$$

Proposition 3. $P(\cdot)$ is continuous at I, and hence $\partial P(I) \neq \emptyset$. In particular, $*(P^*)(I) = P(I)$ and the dual problem $*(P^*)(I)$ has solutions.$\blacksquare$

It is now an easy matter to show that the dual problem actually has solutions in $\tau_s(H)$, that is solutions in $\mathcal{L}_s(H)^*$ with 0 singular part. This leads to the main theorem.

MAIN THEOREM. Let H be a complex Hilbert space and suppose $(f_1,\dots,f_n) \in \tau_s(H)^n$. Then the quantum detection problem has solutions. Moreover, the following statements are equivalent for $m=(m_1,\dots,m_n) \in \mathcal{L}_s(H)^n$:

1) m solves the quantum detection problem

2) $\sum_{j=1}^{n} m_j = I$; $m_i \ge 0$ for $i=1,\dots,n$;

$\sum_{j=1}^{n} f_j m_j \le f_i$ for $i=1,..,n$

3) $\sum_{j=1}^{n} m_j = I$; $m_i \ge 0$ for $i=1,\dots,n$;

$$\sum_{j=1}^{n} m_j f_j \leq f_i \quad \text{for } i=1,\ldots,n.$$

Under any of the above conditions if follows that

$y = \sum_{j=1}^{n} f_j m_j = \sum_{j=1}^{n} m_j f_j$ is self-adjoint and is the unique solution of the dual problem.

4. FINAL REMARKS

Quantum estimation theory is the continuous counterpart of the problem treated here. The main difficulties here are in problem formulation and having an adequate theory of integration with respect to operator-valued measures. For details, see [1] and [6].

REFERENCES

[1] S.K. Mitter and S.K. Young: Quantum Detection and Estimation Theory, I,II, to appear.

[2] H. Yuen, R. Kennedy and M. Lax: Optimum Testing of Multiple Hypotheses in Quantum Detection Theory, IEEE Trans. on Information Theory, IT-21, 1975, p. 25.

[3] A.S. Holevo: Remarks on Optimal Quantum Measurements, Problemy Peredachi Informatsii, Vol. 10, No. 4, 1974, p. 51.

[4] I. Ekeland and R. Temain: Analyse Convexe et Problèmes Variationnels, Dunod-Gauthier-Villars, 1974.

[5] E.B. Davies: Quantum Theory of Open Systems, Academic Press, New York, 1976.

[6] S.K. Young: Doctoral dissertation, M.I.T., August 1977.

CONTINUOUS QUANTUM FILTERING

JOHN S. BARAS
Electrical Engineering Department
University of Maryland
College Park, Maryland 20742

ABSTRACT

We consider here the general filtering problem for signals carried by quantum mechanical fields. The continuous time, continuous quantum measurement problem is formulated. Simple examples from quantum optics illustrate the approach.

1. INTRODUCTION

Communication theory at optical frequencies requires quantum mechanical description of the various relevant devices and fields in order to analyze properly the phenomenon of "quantum noise". This is the type of noise that appears when the carrier frequency is at the optical or higher range of the spectrum, where thermal noise becomes negligible. For a very nice description of "quantum noise" we refer to [5]. Motivated by detection, estimation problems in quantum optics one is lead to the analysis of similar problems for signals carried by quantum fields in general. For a very comprehensive treatment of such problems we refer to [4].

Recently in [1] - [3] the study of the problem of filtering a signal carried by a quantum field was initiated. Due to the enormous complexity and difficulty in the formulation of this problem several simplifying assumptions were made to render the problem feasible. It is the purpose of the present paper to introduce some initial efforts towards a general quantum filtering theory, without the special assumptions of [1] - [3]. We briefly describe the necessary quantum mechanical framework and we refer to [6] for further details. We are only interested in quantum statistical mechanics. Let $\mathcal{H}$ be a complex Hilbert space with inner product $<\cdot , \cdot>$. The state of the quantum system is a self-adjoint, non-negative trace class operator of trace one, usually denoted by ρ. We let $\mathcal{T}_s(\mathcal{H})$ denote the set of all self-adjoint trace class operators on $\mathcal{H}$ with the trace norm

$$||A||_{tr} = tr(A^*A)^{1/2} = \sum_{n=1}^{\infty} \lambda_n \tag{1}$$

where λ_n are the eignevalues of the compact operator $(A^*A)^{1/2}$. This is a Banach space with positive cone $\mathcal{T}_s(\mathcal{H})^+$. Customarily a measurement is represented by a self-adjoint operator on $\mathcal{H}$, V, or with the associated projection - valued measure E_V. If we let v denote the measurement outcome then the probability distribution function of v is given by

$$F_v(\xi) = tr[\rho E_V(-\infty, \xi)] \tag{2}$$

for $\xi \in \mathbb{R}$. Since this concept of a measurement proved to be restrictive for estimation problems [1] - [4] the more general concept of a measurement represented by a positive operator valued measure was introduced. A p.o.m. M is a map from the σ-algebra of Borel sets $\mathcal{B}^n$ of $\mathbb{R}^n$, to the algebra $\mathcal{L}(\mathcal{H})$ of all bounded operators on $\mathcal{H}$, such that

$$\begin{array}{ll} \text{i)} & M(B) \text{ is self-adjoint and } \geq 0 \text{ for every } B \in \mathcal{B}^n \\ \text{ii)} & \text{if } \{B_i\} \subseteq \mathcal{B}^n \text{ is a partition of } \mathbb{R}^n \text{ then } \sum_i M(B_i) = I \text{ (weakly).} \end{array} \tag{3}$$

This a necessary concept for effectively handling vector signal processes [2] - [4]. For more details and relations to Naimark's theorem as well as interpretations in terms of approximate joint measurement of non-compatible variables we refer to [4] and in particular to [7, ch. 3]. The measurement outcome v is in $\mathbb{R}^n$ now, and its statistics are described by the probability measure

$$\mu_v(B) = \mathrm{tr}[\rho M(B)], \text{ for } B \in \mathcal{B}^n. \tag{4}$$

In [1] - [3] we assumed that ρ, the state of the quantum field, depended on a stochastic process x(t), thought as the information carrying signal. To avoid complications, and in particular the time evolution of ρ, it was assumed that ρ does not depend explicitly on time. It was also assumed that $x(t)$ is a discrete time stochastic process. At discrete instants of time t_i, a measurement was performed on the system and gave outcome $v(t_i)$. Both cases where $x(t_i), v(t_i) \in \mathbb{R}$ or $\mathbb{R}^n$ were considered. Again, in order to avoid complications arising from the time evolution of ρ and the well-known interaction between the state of a quantum system and a measurement [6] [7], it was assumed that the measurement outcomes $v(t_i)$ conditioned upon the signal sequence $x(t_i)$ are independent from time to time. This assumption leads to a factorization of the probability distribution function:

$$F_{v(t_1)\cdots v(t_k)\,|\,x(t_1)\cdots x(t_k)} = \prod_{i=1}^{k} F_{v(t_i)\,|\,x(t_i)}\,. \tag{5}$$

A filtered estimator for $x(t_k)$ was formed

$$\hat{x}(t_k) = \sum_{i=0}^{k} C_i(k)\, v(t_i) \tag{6}$$

and the problem of selecting the nxn matrices $C_i(k)$ and p.o.m. M_k in order to minimize the mean square error

$$\text{M.S.E} = E\{\,\| x(t_k) - \hat{x}(t_k) \|_{\mathbb{R}^n}\} \tag{7}$$

was solved. This is an example of a discrete time linear filtering problem with quantum measurements. The results obtained depend critically on the assumptions made above.

In this paper we want to formulate a general filtering problem, without making use of the assumptions used in [1] - [3]. In particular we want to bring into the formulation the time evolution of ρ and the interaction between state propagation in quantum systems and quantum measurements. We shall denote by $\mathcal{L}_s(\mathcal{H})$ the space of all self-adjoint bounded operators on $\mathcal{H}$ with positive cone $\mathcal{L}_s(\mathcal{H})^+$.

2. STOCHASTIC EVOLUTION OF STATES

If H is the Hamiltonian operator of a closed quantum system [7], it is a consequence of the Schrödinger equation that the state $\rho(t)$ evolves according to the evolution equation in $\mathcal{T}_s(\mathcal{H})$:

$$\frac{\partial \rho(t)}{\partial t} = -i[H, \rho(t)] \tag{8}$$

or in integrated form

$$\rho_t = T_t(\rho) = e^{-iHt} \rho e^{iHt} , \tag{9}$$

where we assumed H is time-independent. The interpretation of (8) depends on the particular representation of the Hilbert space $\mathcal{H}$, but usually H is an unbounded self-adjoint operator. T_t in (9) is a strongly continuous group of isometries and for more details on this subject we refer to [7]. Quite often, and, as we shall see shortly, in particular for us, it is necessary to consider equations, like (8) where H depends on a stochastic process x(t). Then (8) becomes a stochastic evolution equation which is quite difficult and yet not known how to handle in this generality. For "nice" states we have shown in [8] that (8) reduces to a stochastic partial differential equation, typically of the Fokker-Planck type, which can be handled quite satisfactorily. In particular we singled out the case where H depends linearly on a stochastic process, since this is frequently the case for common modulation schemes of laser beams [8] [9]. For example electrooptic amplitude modulation of a single mode laser can be modeled by the interaction Hamiltonian

$$H_I = -i g x(t) [a - a^*] , \tag{10}$$

while electrooptic phase modulation can be modeled by

$$H_I = g x(t) a^* a , \tag{11}$$

where a, a^* are the photon annihilation and creation operators of the mode [6], and g a coupling constant. Then the total Hamiltonian of the laser mode is

$$H = \omega a^* a + H_I \tag{12}$$

where ω is the carrier frequency and is clearly linear in the signal process x(t). For stochastic excitations x(t) with piecewise continuously differentiable sample paths equations like (8) can be handled rather satisfactorily whenever H is linear in the process x(t). In particular if

$$H(t) = H + x(t)B \tag{13}$$

where H is self-adjoint, unbounded and generates a unitary group, while $B \in \mathcal{L}_s(H)$ (8) has been studied in [13], with stochastic inputs x(t). This case however exludes hamiltonians arising from Boson quantum fields (such as from lasers), because then the operator B is unbounded. Indeed since a, a^* are unbounded for Boson fields (10) and (11) provide such examples. Fortunately in the most usual cases B is quadratic in a and a^* and then one can still study (8) quite satisfactorily, see [8] and [12] [13]. The same is expected to be true for unbounded operators B which are polynomials in a, a^*. These are the only cases we consider here. Clearly in the case of a vector stochastic signal the term x(t)B in (13) should be replaced by a term $\sum_{i=1}^{n} x_i(t)\, B_i$, where the component processes $x_i(t)$ and operators B_i have similar properties. In general therefore the modulation process (i.e. the injection of the signal to the quantum field) is described by the stochastic evolution equation

$$\frac{\partial \rho(t)}{\partial t} = -i[H, \rho(t)] - i \sum_{i=1}^{n} x_i(t)[B_i, \rho(t)] \tag{14}$$

Via (14) $\rho(t)$ becomes an operator-valued stochastic process.

After modulation the quantum field is transmitted, and therefore interacts with the external world. We model this by a reservoir with hamiltonian H_R and an interaction Hamiltonian gH_I, where g is again a coupling constant. Let $H_\Sigma(t)$ be the Hamiltonian in (14), then the total system is represented on $\mathcal{H} \otimes \mathfrak{F}$ ($\mathfrak{F}$ the Hilbert space of the reservoir) by the total state ρ_1 which satisfies the equation

$$\frac{\partial \rho_1(t)}{\partial t} = -i\,[H_1(t), \rho_1(t)] \tag{15}$$

where

$$H_1(t) = H_\Sigma(t) \otimes I + I \otimes H_R + gH_I \tag{16}$$

Then one employes Master equations methods (i.e. tracing over the reservoir variables) [7] to obtain the stochastic evolution of the state ρ of the quantum field including transmission effects.

3. QUANTUM MEASUREMENT PROCESSES

After modulation and transmission the information carrying quantum field reaches the receiver where measurements are made. Suppose we first make a measurement represented by the self-adjoint operator Y, or the projection valued measure E_Y which is discrete (i.e. Y has discrete spectrum). If y_i is an eigenvalue of Y, it is conventionally supposed that a measurement which gives the value y_i transforms the original state ρ to the state

$$\frac{P_i \rho P_i}{\operatorname{tr}[\rho P_i]} \tag{17}$$

where P_i is the corresponding eigen projection [7]. For convenience one does not consider the normalization factors and constructs the transformation

$$\rho \longmapsto \sum_{i=1}^{\infty} P_i \rho P_i \tag{18}$$

which describes the effect on ρ when Y is measured. This is a conditional expectation in the sense of Umegaki [15] which cannot be generalized however to non discrete observables. It is strongly related to von Neuman's repeatability hypothesis and the related difficulties with continuum spectrum. It is this interaction between states and measurement that was bypassed by the various assumptions in [1] - [3]. In [16] this interaction is considered when measurements are made in discrete instants of time and the underlying quantum states are Gaussian. Here we follow and extend the general approach of Davies [7], [10] - [14] who considered quantum stochastic processes which are the appropriate generalization of marked point-processes. If B is a self-adjoint operator with spectral measure E_B, one defines in the same spirit as (18) the observable conditioned by the measurement of Y by

$$E_{B|Y}(F) = \sum_{i=1}^{\infty} P_i E_B(F) P_i , \text{ for } F \in \mathcal{B} , \tag{19}$$

Note that $E_{B|Y}$ is a p. o. m and not a projection-valued measure. Moreover the joint measurement of Y and B is characterized by the p. o. m

$$E_{Y,B}(F \times E) = \sum_{y_i \in E} P_i E_B(F) P_i . \tag{20}$$

In [14] Davies and Lewis defined the proper generalizations of these ideas by introducing the concept of an instrument. Let $\mathcal{Y}$ be a set and $\mathcal{B}$ a σ-algebra of subsets of $\mathcal{Y}$. An instrument with values in $\mathcal{Y}$ is a map $\mathcal{E}$ from $\mathcal{B}$ into $\mathcal{L}^+(\mathcal{V})$ (the space of bounded positive linear maps of $\mathcal{V}$ into $\mathcal{V}$ where $\mathcal{V} = \mathcal{T}_s(\mathcal{H})$) such that

i) $\mathcal{E}(E) \geq \mathcal{E}(\phi) = 0$, for $E \in \mathcal{B}$

ii) if $\{E_i\} \subseteq \mathcal{B}$ is a countable collection of disjoint sets in $\mathcal{B}$ then

$$\mathcal{E}\Big(\bigcup_{i=1}^{\infty} E_i\Big) = \sum_{i=1}^{\infty} \mathcal{E}(E_i) \quad \text{(strongly.)} \tag{21}$$

iii) $\mathrm{tr}[\mathcal{E}(\mathcal{Y})\rho] = \mathrm{tr}[\rho]$

for all $\rho \in \mathcal{T}_s(\mathcal{H})$.

This corresponds to the idea that an instrument accepts a state ρ and produces a measurement outcome and a new state conditional on the measurement outcome. Typically for us $\mathcal{Y}$ will be either a finite set, or $\mathbb{R}^n$. By employing the fact that $\mathcal{L}_s(\mathcal{H})$ is the dual of $\mathcal{T}_s(\mathcal{H})$ we have [7][14] that to every instrument on $(\mathcal{Y}, \mathcal{B})$ there corresponds a unique p. o. m. M (3) such that

$$\mathrm{tr}[\mathcal{E}(E)\rho] = \mathrm{tr}[M(E)\rho] \text{ for } \rho \in \mathcal{T}_s(\mathcal{H}) . \tag{22}$$

We call M the measurement performed by the instrument $\mathcal{E}$. Conversely given a measurement M on $(\mathcal{Y}, \mathcal{B})$, choose a partition $\{E_i\}$ of $\mathcal{Y}$ and a sequence of states ρ_i, with $\mathrm{tr}[\rho_i] = 1$. Then the formula

$$\mathcal{E}(E)\rho = \sum_{i=1}^{\infty} \mathrm{tr}[M(E \cap E_i)\rho]\rho_i \tag{23}$$

defines an instrument on $(\mathcal{Y}, \mathcal{B})$ which performs the measurement M. Suppose $\mathcal{E}^1, \mathcal{E}^2$ are instruments on $(\mathcal{Y}^1, \mathcal{B}^1)$, $(\mathcal{Y}^2, \mathcal{B}^2)$ with $\mathcal{Y}^1, \mathcal{Y}^2$ complete separable metric spaces, then there is a unique instrument $\mathcal{E}^{1,2}$, called the composition of $\mathcal{E}^2$ following $\mathcal{E}^1$, on $\mathcal{Y}^1 \times \mathcal{Y}^2$ such that

$$\mathcal{E}^{1,2}(E_1 \times E_2)\rho = \mathcal{E}^2(E_2)\mathcal{E}^1(E_1)\rho \tag{24}$$

for all $E_1 \in \mathcal{B}^1$, $E_2 \in \mathcal{B}^2$, $\rho \in \mathcal{T}_s(\mathcal{H})$. Then if M_1, M_2 are the measurements performed by $\mathcal{E}^1, \mathcal{E}^2$, the p. o. m.

$$M_{2|1}(F) = \mathcal{E}^1(\mathcal{Y}^1)^* M_2(F), \quad F \in \mathcal{B}^2 \qquad (25)$$

is the measurement M_2 conditioned by the measurement M_1. Moreover the joint statistics of the two measurements are given by the probability measure

$$\mu_{1,2}(A) = \mathrm{tr}[\mathcal{E}^{1,2}(A)\rho] \,/\, \mathrm{tr}[\rho], \quad A \in \mathcal{B}^1 \times \mathcal{B}^2. \qquad (26)$$

For our purposes we need a more general concept. Namely a one parameter family of instruments: a "quantum process". Davies in [7] and [10-12] developed such a concept, but for measurements with outcomes forming a point process. As a generalization we introduce a quantum measurement process. Let $\mathcal{Y}$ a complete separable metric space and $\mathcal{B}$ the Borel σ-algebra of $\mathcal{Y}$. Let $\mathcal{Y}^t$ be the set of all measurable functions from $[0, t]$ into $\mathcal{Y}$ with the usual σ-algebra. Given $s, t \geq 0$ there is a one-one Borel map c of $\mathcal{Y}^s \times \mathcal{Y}^t$ onto $\mathcal{Y}^{s+t}$ defined by concatenation

$$c(\omega^s, \omega^t)(\tau) = (\omega^s \circ \omega^t)(\tau) = \begin{cases} \omega^s(\tau) & , 0 \leq \tau < s \\ \omega^t(\tau - s), & s \leq \tau \leq t+s \end{cases} . \qquad (27)$$

A quantum measurement process is a family of instruments $\mathcal{E}^t$ on $\mathcal{Y}^t$ for $t \geq 0$ such that

i) $\lim\limits_{t \geq 0} \mathcal{E}^t(\mathcal{Y}^t)\rho = \rho$, for $\rho \in \mathcal{T}_s(\mathcal{H})$

ii) for $\rho \in \mathcal{T}_s(\mathcal{H})$ and $s, t \geq 0$

$$\mathcal{E}^t(F)\mathcal{E}^s(E)\,\rho = \mathcal{E}^{s+t}(c(E \times F))\,\rho \qquad (28)$$

for $F \in \sigma$-algebra of $\mathcal{Y}^t$, $E \in \sigma$-algebra of $\mathcal{Y}^s$

According to the desirable properties of the measurement outcome process, additional assumptions on $\mathcal{Y}^t$ can be made which will give (28) more structure (such as : continuous, square integrable, mark point processes). In this generality we do not know as yet how to characterize the process. Davies has studied extensively such processes when $\mathcal{Y}^t = \{(y_i, t_i) : 1 \leq i \leq n,$ n arbitrary$\}$. If we let z denote the empty sample path, he analyzed processes with bounded interaction rate in the sense that

$$\mathrm{tr}[\mathcal{E}^t(\mathcal{Y}^t - \{z\})\rho] \leq K t \,\mathrm{tr}[\rho]. \qquad (29)$$

Defining the two semigroups on $\mathcal{T}_s(\mathcal{H})$

$$T_t(\rho) = \mathcal{E}^t(\mathcal{Y}^t)\rho \qquad (30)$$

$$S_t(\rho) = \mathcal{E}^t(\{z\})\rho \qquad (31)$$

and assuming that S_t has the form

$$S_t(\rho) = B_t \rho B_t^* \qquad (32)$$

where B_t a strongly continuous contraction semigroup on $\mathcal{H}$, be showed that they are uniquely characterized by the infinitesimal generator Ψ of B_t and a positive map valued measure on $\mathcal{Y}, \mathcal{J}$ (i.e. which satisfies (21) with iii) replaced by $\mathrm{tr}[\mathcal{J}(\mathcal{Y})\rho) \leq K\,\mathrm{tr}[\rho]$) and are related by:

$$\mathrm{tr}[\mathcal{J}(\mathcal{Y})\,|b\rangle\langle b|] = -2\,\mathrm{Re}\langle \Psi b, b\rangle \tag{33}$$

for all $b \in \mathcal{D}(\Psi)$. An important role in the anlysis is played by the total interaction rate of the process R defined by

$$\mathrm{tr}[R\rho] = \mathrm{tr}[\mathcal{J}(\mathcal{Y})\rho] \quad . \tag{34}$$

Then it can be shown that if H(t) is the Hamiltonian of the quantum system before interacting with measurements the state obeys the differential equation

$$\begin{aligned}\frac{\partial \rho(t)}{\partial t} &= -i[H(t), \rho] + \mathcal{J}(\mathcal{Y})\rho \\ &\quad - \frac{1}{2}\{R\rho + \rho R\}\end{aligned} \tag{35}$$

see [7] [13] for details. This has also been established for Boson fields with photon counters as measurements in [12] [13], which are processes with unbounded interaction rates.

4. QUANTUM FILTERING

We are now ready to formulate the general quantum filtering problem. Let $\rho(t, x^t)$ be the state of the quantum system at time t, which depends on the sample path x^t of the signal up to time t. This is obtained from sections 2 and 3 above assuming a "modulating" interaction Hamiltonian linear in x(t). More complicated dependence, including memory is clearly possible. Given a class C of quantum measurement processes we have available from each the classical process y(t) which denotes the measurement out comes. The general filtering problem is: Choose the quantum measurement process $\mathcal{E}^t$ from C and functionals f_t measurable with respect to $\sigma(y_s, s \leq t)$ such that

$$\hat{x}(t) = f_t(y_s, s \leq t) \tag{36}$$

is the minimum variance estimator of x(t).

Motivated by the work of Davies [13] we consider the question: Does there exist a family of quantum measurement processes in C, parametrized by the sample paths x^t of the signal process on $\mathcal{Y}^t$

$$\mathcal{E}^t(x^t, E)\rho(o) \tag{37}$$

such that

$$\rho(t) = \mathcal{E}^t(x^t, \mathcal{Y}^t)\rho(o) \tag{38}$$

satisfies an equation like (35)? In what sense? Once this is established further progress can be made, since then the probability measure

$$\mu_t(x^t, E) = \mathrm{tr}[\mathcal{E}^t(x^t, E)\rho(o)] \tag{39}$$

characterizes the statistics of the classical measurement process y(t) given the classical signal process x(t). We now give two simple examples to illustrate the approach.

The first is similar to that of Davies [13]. We consider an amplitude modulated single mode laser with the class C consisting of photon counters. The Hamiltonian before measurement is given by (10) (12). We let $\mathcal{H}$ be the Boson Fock space for a single mode [7] [8]. The coherent states $|\alpha\rangle\langle\alpha|$ with $|\alpha\rangle$ an eigenvector of the annihilation operator a play an important role in the sequel since they provide good models for lasers above threshold and their linear span is dense in $\mathcal{T}_s(\mathcal{H})$. Here $\mathcal{Y}$ collapses to a single point. If we start at a state $|\alpha\rangle\langle\alpha|$ a measurement of a photon will reduce the number of photons by 1 and therefore the new state will be a $|\alpha\rangle\langle\alpha| a^*$. So the instrument is described by

$$\mathcal{E}\,\rho = a\,\rho\,a^* = \mathcal{J}\rho. \tag{40}$$

From (34) $R = a^* a$. So (35) gives

$$\begin{aligned}\frac{\partial\rho(t)}{\partial t} = &- i\,\omega[a^* a, \rho(t)] - g\,x(t)\,[a - a^*, \rho(t)] \\ &- \frac{1}{2}\,[a^* a\,\rho(t) + \rho(t)\,a^* a] + a\,\rho(t)\,a^*\end{aligned} \tag{41}$$

If we work with classical states, i.e. $\rho = \int |\alpha\rangle\langle\alpha|\, P(\alpha)\, d^2\alpha$ (42)

we can translate (41) into a p. d. e. for P [8]. We obtain

$$\begin{aligned}\frac{\partial P(\alpha, t)}{\partial t} = &- i\omega\left(\frac{\partial}{\partial\overline{\alpha}}\,\overline{\alpha} - \frac{\partial}{\partial\alpha}\,\alpha\right) P(\alpha, t) \\ &- g\,x(t)\left(\frac{\partial}{\partial\alpha} + \frac{\partial}{\partial\overline{\alpha}}\right) P(\alpha, t) \\ &- \frac{1}{2}\left[2\,\overline{\alpha}\,\alpha - \frac{\partial}{\partial\alpha}\,\alpha - \frac{\partial}{\partial\overline{\alpha}}\,\overline{\alpha}\right] P(\alpha, t) + \alpha\overline{\alpha}\, P(\alpha, t)\,,\end{aligned} \tag{43}$$

or if $\alpha = \zeta + i\xi$ we have the Fokker-Planck equation

$$\begin{aligned}\frac{\partial P(\zeta, \xi, t)}{\partial t} = &- \omega\left(\xi\,\frac{\partial}{\partial\zeta} - \zeta\,\frac{\partial}{\partial\xi}\right) P(\zeta, \xi, t) \\ &- g\,x(t)\,\frac{\partial}{\partial\zeta}\, P(\zeta, \xi, t) \\ &+ \frac{1}{2}\left(\zeta\,\frac{\partial}{\partial\zeta} + \xi\,\frac{\partial}{\partial\xi} + 2\right) P(\zeta, \xi, t).\end{aligned} \tag{44}$$

This is easily seen to correspond to the linear stochastic differential equation:

$$\begin{bmatrix} d\,\zeta(t) \\ d\,\xi(t) \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} & \omega \\ -\omega & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} \zeta(t) \\ \xi(t) \end{bmatrix} + g\,x(t) \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{45}$$

Again since (44) is a degenerate Fokker-Planck equation we see in the same fashion as in [8] that if we start at a coherent state

$$\rho(0) = |\alpha_0 >< \alpha_0|$$

we will end at the coherent state

$$\rho(t) = |\alpha(t) >< \alpha(t)| \tag{46}$$

where $\alpha(t) = \zeta(t) + i\xi(t)$ is computed from (45) or from

$$\frac{d\alpha(t)}{dt} = \left(-i\omega - \frac{1}{2}\right)\alpha(t) + g\,x(t) \tag{47}$$

$$\alpha(0) = \alpha_o .$$

Now in this case the classical measurement process is a point process $\mathcal{Y}(t)$. Let N_t denote the corresponding counting process [18]. The most relevant quantity is the rate process

$$\begin{aligned}&\lambda_t(N_t; w_1; \ldots; w_{N_t}; x(s),\ s \leq t) = \\ &= \lim_{\Delta t \to 0} \Delta t^{-1}\ \Pr(N_{t,\,t+\Delta t} = 1 \mid N_t; w_1; \ldots; w_{N_t}; x^t)\end{aligned} \tag{48}$$

where $w_1, \ldots w_{N_t}$ are the occurence times. It can be seen through the construction of ${}^t\mathcal{E}^t$ [13] that in this case

$$\begin{aligned}\lambda_t &= \mathrm{tr}[\mathcal{J}\rho(t)] = \mathrm{tr}\,[a|\alpha(t) >< \alpha(t)|a^*] \\ &= |\alpha(t)|^2 = \left|g \int_{s=0}^{t} e^{-(i\omega + \frac{1}{2})(t-s)} x(s)ds\right|^2\end{aligned} \tag{49}$$

This has now been reduced to a classical problem which can be solved. Notice that no measurement optimization was involved here.

In the second example we consider a monochromatic laser beam operating above threshold which carries a two dimensional real signal as its in phase and quadrature amplitudes $x_1(t)$, $x_2(t)$. The received field's state has a P-representation with

$$P(\alpha, t) = \frac{1}{\pi \bar{n}} \exp\left\{-\frac{|\alpha - x_1(t) + i x_2(t)|^2}{\bar{n}}\right\} \tag{50}$$

[2]. We assume the signals are generated according to a linear stochastic differential equation

$$\begin{bmatrix} \dfrac{d\,x_1(t)}{dt} \\ \dfrac{d\,x_2(t)}{dt} \end{bmatrix} = A(t) \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} dt + B(t) \begin{bmatrix} dw_1(t) \\ dw_2 t \end{bmatrix} \tag{51}$$

where $w_1(t)$, $w_2(t)$ are Wiener processes. We conjecture that the optimal quantum measurement process is one which performs the measurement

$$M(A) = \frac{1}{\pi} \int_A |\alpha >< \alpha|\, d\alpha$$

for all times. Then the classical measurement process can be represented by

$$y(t) = x(t) + v(t) \tag{52}$$

where v(t) is a white noise process with mean zero and variance $\frac{\bar{n}+1}{2}$. (51) and (52) define a classical filtering problem.

REFERENCES

[1] J.S. Baras, R.O. Harger and Y.H. Park, "Quantum Mechanical Linear Filtering of Random Signal Sequences", IEEE Trans. on Info. Th. Vol. IT-22, No. 1, Jan 1976, pp. 59-64.

[2] J.S. Baras and R.O. Harger, "Quantum Mechanical Linear Filtering of Vector Signal Processes", to appear in IEEE Trans on Info. Th., Nov. 1977.

[3] J.S. Baras, "An Optimization Problem from Linear Filtering with Quantum Measurements" to appear

[4] A.S. Holevo, "Statistical Decision Theory for Quantum Systems", J. of Multivariate Analysis, Vol. 3, pp. 337-394.

[5] D. Marcuse, Engineering Quantum Electrodynamics, Harcourt Brace and World, 1970.

[6] E. Prugovecki, Quantum Mechanics in Hilbert Space, Academic Press, 1971.

[7] E.B. Davies, Quantum Theory of Open Systems, Academic Press, 1976.

[8] J.S. Baras, "Stochastic Partial Differential Equations in Optical Communication Problems", Proceed. of III U.S. - Italy Conference on Variable Structure Systems, Taormira, Sicily, Aug. 1977, to be published.

[9] A. Yariv, Quantum Electronics, John Wiley 1967.

[10] E.B. Davies, "Quantum Stochastic Processes", Comm. Math. Phys. Vol. 15, 1969, pp. 277-304.

[11] E.B. Davies, "Quantum Stochastic Processes II", Comm. Math. Phys. Vol. 19, 1970, pp. 83-105.

[12] E.B. Davies, "Quantum Stochastic Processes III", Comm. Math Phys. Vol. 22, 1971, pp. 51-70.

[13] E.B. Davies, "Quantum Communication Systems", IEEE Trans. on Info. Th., July 1977, pp. 530-534.

[14] E.B. Davies and J.T. Lewis, "An Operational Approach to Quantum Probability", Comm. Math. Phys., Vol. 17, 1970, pp. 239-260.

[15] H. Umegaki, "Conditional Expectation in an Operator Algebra", Tohôku Math. J. 6, 1954, pp. 177-181.

[16] J.S. Baras, R.O. Harger and L. Wilhelm, "Signal Filtering with Gaussian Quantum States and Canonical Measurements", to appear in Proc. of 1977 IEEE Intern. Symp. on Info. Th. October 1977.

[17] E.B. Davies, "On the Repeated Measurement of Continuous Observables in Quantum Mechanics", J. of Funct. Analysis, Vol.6, 1970, pp. 318-346.

[18] D.L. Snyder, "Random Point Processes", John Wiley, 1975.

ACKNOWLEDGEMENT

This work was partially supported by the National Science Foundation under grant ENG 75-20900.

ON MODEL REFERENCE ADAPTIVE CONTROL SYSTEMS

R. V. MONOPOLI
Electrical and Computer Engineering Department
University of Massachusetts
Amherst, MA 01003

ABSTRACT

Some unjustified conclusions regarding convergence properties of a class of model reference adaptive control systems have appeared in the literature. The systems are those using only input and output signals to implement the adaptive algorithm. In this paper, it is pointed out when and why the conclusions are unjustified. Also, a discrete adaptive algorithm is proposed which does seem to have the desired convergence properties.

I. INTRODUCTION AND PROBLEM STATEMENT

In model reference adaptive control, the output x(t) of the plant, given by

$$D_p(p)x(t) = b_o D_u(p)u(t) \tag{1}$$

must be made to follow the output $x_m(t)$ of a model given by

$$D_m(p)x_m(t) = K_o r(t) \tag{2}$$

In (1) and (2), $p = d/dt$, $D_p(p) = p^n + a_1 p^{n-1} + \ldots a_n$, $D_u(p) = p^m + b_1 p^{m-1} + \ldots b_m$, $D_m(p) = p^n + a_{d1} p^{n-1} + \ldots a_{dn}$, u(t) is the plant input, r(t) is the system reference input, and $m \leq n - 1$. Plant parameters a_i and b_i are unknown and assumed to be constant. It is also assumed that roots of $D_u(s)$ are in the left half s plane and that the sign of b_o is known (assumed positive).

Parks [1] suggested a parameter adjusting model reference adaptive control system as a solution to this problem. This method was based on Liapunov's direct method. Following Parks' work, others [2]-[7] have made his approach more practical by reducing the number of derivatives of the plant output required. The results reported appear to be theoretically sound except for certain conclusions in [6] and [8] which are not justified. Feuer, et al. [9] seem to have been the first to point out the problem with the conclusion reached in [6]. This problem and that in [8] are examined in the next section. In Section III, the discrete adaptive algorithm reported in [10] is discussed. It appears to solve the problem under consideration when applied to the sampled-data system derived from (1).

II. JUSTIFIED AND UNJUSTIFIED CONCLUSIONS

First, results which can be justified are given. Next, reasons why solutions proposed in [6] and [8] cannot be justified are examined.

An equation for the true system error, $e = x_m - x$, is obtained by subtracting (1) from (2)

$$D_m(p)e(t) = K_o r(t) - b_o D_u(p)u(t) + D_\Delta(p)x(t) \tag{3}$$

where $D_\Delta(p) = D_p(p) - D_m(p)$.

The following statements summarize the justifiable results reported pertaining to the design of a parameter adaptive control for (3):

1. A parameter adaptive control exists for (3) which assures that $e(t) \to 0$ as $t \to \infty$ and which can be implemented without derivatives of the plant output, x, if $n - 2 \leq m \leq n - 1$, and which requires no more than $n - m - 2$ derivatives of x if $0 \leq m < n - 2$, [1]-[5].

2. In the special case where the plant gain b_o is the only unknown parameter, a parameter adaptive control exists for (3) which assures $e(t) \to 0$ as $t \to \infty$ and which can be implemented without derivatives of the plant output x for any m satisfying $0 \leq m \leq n - 1$, [6].

The unjustified claim in [6] appears in connection with the general problem, (V-A), where it is assumed that a_i and b_i, (as well as b_o) are unknown. In this case, the augmented error approach, which yields result 2 above, has not been shown to assure convergence of e to zero. The reason for this is examined next.

The augmented error signal is

$$\eta(t) = e(t) + y(t) \tag{4}$$

where

$$D_m(p)y = D_w(p)w(t) \tag{5}$$

In (5), $w(t)$ is an auxiliary system input to be determined along with u, $D_w(p) = p^{n-1} + c_1 p^{n-2} + \ldots c_n$, and $D_w(s)/D_m(s)$ is strictly positive real (assumed from here on, without loss of generality, to be $(s + \alpha)^{-1}$). The differential equation for η is

$$(p + \alpha)\eta(t) = -b_o D_u(p)z_o + K_o z_1 + D_\Delta(p)z_2 + W \tag{6}$$

where $z_o = D_w^{-1}(p)u$, $z_1 = D_w^{-1}(p)r$ and $z_2 = D_w^{-1}(p)x$. As in [6], (6) can be put in the form

$$(p + \alpha)\eta(t) = -b_o u_f + w + \sum_{i=0}^{2n-1} \beta_i \phi_i \tag{7}$$

where $u_f = D_f^{-1}(p)u$, $D_f(p) = p^{n-m-1} + F_1 p^{n-m-2} + \ldots F_{n-m-1}$, β_i are unknown constants related to plant parameters, and $\phi_o = z_1$, $\phi_i = p^{i-1}D_f^{-1}(p)z_o$ for $i = 1$ to $n - 1$, and $\phi_{n-1+i} = p^{i-1}D_f^{-1}(p)z_2$ for $i = 1$ to n. Let

$$w = k_{2n}w_1 = k_{2n}(t)\phi_{2n}(t) \tag{8}$$

and

$$u_f = \sum_{i=0}^{2n-1} k_i\phi_i - \phi_{2n} \tag{9}$$

Then substitute (8) and (9) into (7) to obtain

$$(p + \alpha)\eta(t) = \sum_{0}^{2n} \delta_i(t)\phi_i(t) \tag{10}$$

where $\delta_i(t) = -b_o k_i(t) + \beta_i$ for $i = 0$ to $2n - 1$, and $\delta_{2n}(t) = k_{2n} + b_o$.

The Liapunov function for (10)

$$V = (1/2)[\eta^2 + \sum_{i=0}^{2n} \lambda_i \delta_i^2] \tag{11}$$

(where λ_i are arbitrary positive constants), has time derivative

$$\dot{V} = -\alpha\eta^2 \tag{12}$$

if the parameter adjusting algorithm is chosen to be

$$\begin{aligned} \dot{k}_i &= A_i \eta \phi_i && \text{for } i = 0 \text{ to } 2n - 1 \\ \dot{k}_{2n} &= -A_{2n}\eta\phi_{2n} && (A_i = \text{pos. const. for } i = 0 \text{ to } 2n) \end{aligned} \tag{13}$$

The plant input u and part of the auxiliary input, w_1, are found by applying the operator $D_f(p)$ to (9) to obtain

$$u = \sum_{i=0}^{2n-1} k_i D_f \phi_i \tag{14}$$

$$w_1 = D_f^{-1}(p)[\sum_{i=0}^{2n-1} \sum_{j=0}^{n-m-2} D_{fj}(p)((pk_i)(p^{n-m-2-j}\phi_i))] \tag{15}$$

where $D_{f0}(p) = 1$, $D_{f1}(p) = p + F_1$, ... , and
$D_{f(n-m-2)}(p) = p^{n-m-2} + F_1 p^{n-m-3} + \ldots F_{n-m-2}$.

The conclusion reached in [6] is that the adaptive system defined by (13)-(15) assures that $e \to 0$. This conclusion is based on the argument that (12) and (11) imply that $\eta \to 0$ and that $\eta \to 0$ implies that $w_1 \to 0$ because the inputs to the stable linear filters generating w_1, i.e., $(pk_i)(p^{n-m-2-j}\phi_i)$, contain η as a factor and therefore approach zero. This is clearly not the case in general. For example, one input to the filters generating w_1 is $\eta\phi_i^2$ and it may not go to zero if $\phi_i^2 \to \infty$ faster than $\eta \to 0$. Thus w_1 may not approach zero. In this case, $\eta \to 0$ will only imply that $e \to -y$, but it may not approach zero. In the special case where b_0 is the only unknown plant parameter, the conclusion that $e \to 0$ is valid because ϕ_0 is the only one of the ϕ_i required in (14) and (15) and it will be bounded since it is the output of a stable linear system driven by r (which is assumed bounded). In the general case, the conclusion that $e \to 0$ cannot be justified by the arguments presented. However, simulation results by the author and others [7] have shown that the algorithm works at least in some cases including even unstable plants [7]. In fact, in [7], Narendra and Valevani conjecture that their approach (similar to that in [6]) will work in general in spite of the lack of a rigorous theoretical proof to this effect. No counter examples to the conjecture have been reported. However, Wolovich [11] feels that the schemes proposed which rely on only a scalar error signal will not work if the plant is "sufficiently" unstable because in the time it takes an adaptive controller to adjust n or more system parameters, the output of an unstable plant could become "practically unbounded".

In [8], a solution is proposed which is based on representing the plant in the Lüders-Narendra canonical form [12], i.e.,

$$\text{Plant:}\quad \dot{x}_1 = a_1x_1 + \sum_{i=2}^{n} x_1 + b_1u$$

$$\dot{x}_k = a_kx_1 - \lambda_kx_k + b_ku \qquad k = 2, \ldots , n$$

$$y = x_1 \tag{16}$$

where λ_k are arbitrary distant positive numbers and a_i and b_i are unknown plant parameters.

It is the case, though not stated in [8], that this proposed solution is only valid for systems in which $b_1 \neq 0$, i.e., systems for which $m = n - 1$, a case which is included under item 1 of the justifiable results. Therefore, the conclusion that this technique will work in the general case is not justified.

III. A DISCRETE SOLUTION

Although claims for the adaptive algorithm in the continuous case are not justified, a solution seems possible if the discrete algorithm reported in [10] is applied to the sampled data equivalent of (1). In this approach, the discrete time output x(k) of the plant described by

$$Z_p(n)x(k + n) = b_0Z_u(n - 1)u(k + n - 1) \tag{17}$$

must follow the output of a model given by

$$Z_M(n)x_m(k + n) = K_0r(k) \tag{18}$$

In (17) and (18) Z_p, Z_u, and Z_M are polynomials in the delay operator z^{-1}. The details of the general case, which appear in [10], will not be repeated here. Instead, the method is illustrated by considering the following example. The plant, model, and augmenting network equations

$$\text{Plant:}\quad x(k + 1) + ax(k) = b_0u(k) \tag{19}$$

$$\text{Model:}\quad x_m(k + 1) + \alpha x(k) = K_0r(k) \tag{20}$$

$$\text{Augmenting network:}\quad y(k + 1) + \alpha y(k) = q(k + 1) \tag{21}$$

can be manipulated to give the augmented error equation

$$\eta(k + 1) + \alpha\eta(k) = K_0r(k) - b_0u(k) + \Delta ax(k) + q(k + 1) \tag{22}$$

where $\eta(k) = e(k) + y(k)$, $e(k) = x_m(k) - x(k)$, a and b_o are unknown, $|\alpha| < 1$, and u and q are to be synthesized such that $e(k) \to 0$ as $k \to \infty$. It must be assumed that $0 \leq b_{om} \leq b_o \leq b_{oM} < \infty$ and b_{om} and b_{oM} are known.

The control signal is chosen to be

$$u(k) = K_1(k)\phi_1(k) + K_2(k)\phi_2(k) \tag{23}$$

where $\phi_1(k) = r(k)$ and $\phi_2(k) = x(k)$ and $K_i(k)$ are adjustable gains. Substituting (23) in (22) gives

$$\eta(k + 1) + \alpha\eta(k) = \sum_{1}^{2} \delta_i(k)\phi_i(k) + q(k + 1)$$

$$= \Delta(k) + q(k + 1) = \xi(k) \tag{24}$$

where $\delta_1(k) = -b_oK_1(k) + K_0$, $\delta_2(k) = -b_oK_2(k) + \Delta a$.

It is claimed that (23) and the following gain adjusting algorithm and auxiliary input q will insure $e(k) \to 0$ as $k \to \infty$:

$$K_i(k) - K_i(k - 1) = -(\lambda b_o)^{-1}K_K(k)\eta(k)\phi_i(k - 1), \quad i = 1, 2 \tag{25}$$

and

$$q(k) = -K_Q(k)\eta(k) \tag{26}$$

where $K_Q(k) = K_q\Sigma_1(k) = K_q[K_k + \Sigma(k)] = K_q[K_k + \sum_{i=1}^{2} \phi_i^2(k - 1)]$, $K_K(k) = K_k(\Sigma_1(k))^{-1}$, λ is an arbitrary positive constant and K_q and K_k are positive constants satisfying

$$K_qK_k > 2^{-1}[\lambda^{-1} + 1] \tag{27}$$

The following Liapunov function is used to justify this claim:

$$V(k) = (\eta(k) - \xi(k - 1))^2 + \varepsilon\underline{\psi}^T(k - 1)M(k - 1)\underline{\psi}(k - 1)$$

$$+ \sum_{i=1}^{2} \lambda\delta_i^2(k - 1) \tag{28}$$

In (28), $0 < \varepsilon << 1$, $\underline{\psi}^T(k) = [\Delta(k - 1), q(k)]$, and, if (27) is satisfied, M(k) is a positive definite matrix with elements $m_{11} = 1$, $m_{12}(k) = 1 + (1 - K_K(k))K_Q^{-1}$, and $m_{22}(k) = 1 + (2K_Q - \lambda^{-1}K_K^2(k)\Sigma(k))K_Q^{-2}$. For the Liapunov function (28), $\Delta V(k)$ evaluated for (24) satisfies

$$\Delta V(k) < -(1 - \alpha^2)\eta^2(k) + \varepsilon\underline{\psi}^T(k)M(k)\underline{\psi}(k) + 2\eta(k)\xi(k - 1)$$

$$- \xi^2(k - 1) + \sum_{i=1}^{2} \lambda(\delta_i^2(k) - \delta_i^2(k - 1)) \tag{29}$$

Using (25) and (26) in (29), it can be shown that

$$\Delta V(k) < -(1-\alpha^2)\eta(k) - (1-\varepsilon)\underline{\psi}^T(k)M(k)\underline{\psi}(k) \qquad (30)$$

Since V(k) is positive definite in η, Δ, q, and δ and $\Delta V(k)$ neg. def. in η, Δ, and q, then η, Δ, and q all approach zero as $k \to \infty$. The augmenting signal $y \to 0$ since $q \to 0$. Since $\eta \to 0$ and $y \to 0$, $e \to 0$ as required.

IV. CONCLUSIONS

Although certain conclusions appearing in the literature regarding adaptive control algorithms for continuous systems are incorrect, the problem they attempt to solve appears amenable to solution via a discrete adaptive algorithm applied to the sampled-data version of the continuous system.

ACKNOWLEDGEMENT

Research reported herein was partially supported by the National Science Foundation under grant ENG-74-03399.

REFERENCES

[1] P. C. Parks, "Liapunov Redesign of Model Reference Adaptive Control Systems," IEEE Trans. Automat. Contr., Vol. AC-11, pp. 362-367, July 1966.

[2] R. V. Monopoli, "Liapunov's Method for Adaptive Control-System Design," IEEE Trans. Automat. Contr. (Corresp.), Vol. AC-12, pp. 334-335, June 1967.

[3] J. W. Gilbart, R. V. Monopoli, and C. F. Price, "Improved Convergence and Increased Flexibility in the Design of Model Reference Adaptive Control Systems," in Proc. 1970 IEEE Symp. Adaptive Processes (9th), Decision and Control, Austin, TX.

[4] R. V. Monopoli, "Adaptive Control and Identification via Liapunov's Direct Method," in Proc. Allerton Conf., University of Illinois, Urbana, 1972.

[5] R. V. Monopoli, "The Kalman-Yacubovich Lemma in Adaptive Control System Design," IEEE Trans. on Automat. Contr., Vol. AC-18, No. 5, pp. 527-529, October 1973.

[6] R. V. Monopoli, "Model Reference Adaptive Control with an Augmented Error Signal," IEEE Trans. Automat. Contr., Vol. AC-19, No. 5, October 1974, pp. 474-484.

[7] K. S. Narendra and L. S. Valavani, "Stable Adaptive Controller Design, Part I: Direct Control," S&IS Report No. 7706, June 1977, Department of Engr. and Applied Science, Yale University, New Haven, Connecticut.

[8] K. R. Krishnan, "Adaptive Model-Reference Control of an Unknown Plant with Limited Knowledge of Model," IEEE Trans. on Automat. Contr., Vol. AC-22, No. 3, pp. 484-486, June 1977.

[9] A. Feuer, B. R. Barmish, and A. S. Morse, "An Unstable Dynamical System Associated with Model Reference Adaptive Control," S&IS Report No. 7702, May 1977, Department of Engr. and Applied Science, Yale University, New Haven, Connecticut.

[10] T. Ionescu and R. V. Monopoli, "Discrete Model Reference Adaptive Control with an Augmented Error Signal," to appear in the Automatica, September 1977 issue.

[11] W. Wolovich, Division of Engr., Brown University, Providence, RI, private communication with author.

[12] G. Lüders and K. S. Narendra, "An Adaptive Observer and Identifier for a Linear System," IEEE Trans. Automat. Contr., Vol. AC-18, pp. 496-499, October 1973.

MULTIPLE TIME SCALES IN SINGLE INPUT-SINGLE OUTPUT HIGH GAIN FEEDBACK SYSTEMS*

KAR-KEUNG D. YOUNG
Decision and Control Laboratory
Coordinated Science Laboratory
University of Illinois
Urbana, Illinois 61801

ABSTRACT

The asymptotic behavior of single input-single output high gain feedback systems as the loop gain tends to infinity is investigated by singular perturbation methods. It is found that this class of feedback systems may have different time scales depending on system structure. A decomposition of the computation of the closed loop eigenvalues into two lower order problems is presented. An illustrative example is given.

1. INTRODUCTION

In this paper, we study the asymptotic behavior of single input-single-output feedback systems when the loop gain tends to infinity. While the well known root locus method [1] provides graphical estimates of the closed loop eigenvalues as a function of the loop gain given the transfer function of the open loop system, we analyze the high gain feedback system as a singularly perturbed system and decompose the computation of the closed loop eigenvalues into two separate low order problems.

2. TIME SCALE DECOMPOSITIONS

The class of single input-single output linear time-invariant high gain feedback systems considered is of the form

$$\dot{x} = Ax + bu \tag{1}$$

$$y = c^T x \tag{2}$$

$$u = gy \tag{3}$$

where $x \in R^n$, $u \in R$, $y \in R$ and g is a large positive scalar gain factor. We assume that this system is controllable and introduce the transformation

$$x' = Q^{-1} x \tag{4}$$

where

$$Q = [b \ Ab \ A^2 b \ \ldots \ A^{n-1} b] \tag{5}$$

is the controllability matrix. In the new coordinates, (1), (2) is

---

[*] This work was supported in part by the Joint Services Electronics Program (U.S. Army, U.S. Navy, and U.S. Air Force) under Contract DAAB-07-72-C-0259, in part by the U.S. Air Force under Grant AFOSR-73-2570 and in part by the National Science Foundation under Grant ENG-74-20091.

represented by the system

$$\dot{x}' = \left[\begin{array}{cccc|c} 0 & 0 & \dots & 0 & -\alpha_n \\ \hline & & & & -\alpha_{n-1} \\ & I_{n-1} & & & \vdots \\ & & & & -\alpha_1 \end{array}\right] x' + \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} u \tag{6}$$

and

$$y = [c^T b, c^T Ab, c^T A^2 b, \dots, c^T A^{n-1} b] x'. \tag{7}$$

By defining $\gamma_i = c^T A^{i-1} b$, $i = 1,\dots,n$, the high gain feedback system (1)-(3) has the form

$$\dot{x}' = \left[\begin{array}{cccc|c} g\gamma_1 & g\gamma_2 & \dots & g\gamma_{n-1} & g\gamma_n - \alpha_n \\ \hline & & & & -\alpha_{n-1} \\ & I_{n-1} & & & \vdots \\ & & & & -\alpha_1 \end{array}\right] x'. \tag{8}$$

It is shown in [2] that if $\gamma_1 \neq 0$, then (8) has n-1 eigenvalues which are $O(1)$ and one $O(g)$ large eigenvalue. In this case, the "fast" motion of (8), that is, the motion which can be analyzed in the "stretched time scale" gt, occurs in a one dimensional subspace spanned by the input vector b. Hence, in this paper, we only consider the cases when $\gamma_1 = 0$. Before we present the most general case, we first investigate the case when $\gamma_2 \neq 0$. System (8) then has the form

$$\dot{z}_2 = P_2 z_2 + M_2 w_2 \tag{9}$$

$$w_2 = R_2 z_2 + S_2 w_2 \tag{10}$$

where

$$(z_2)^T = (x_3', x_4', \dots, x_n') \in R^{n-2}, \quad (w_2)^T = (x_1', x_2') \in R^2$$

$$P_2 = \left[\begin{array}{cccc|c} 0 & 0 & \dots & 0 & -\alpha_{n-2} \\ \hline & & & & -\alpha_{n-3} \\ & I_{n-2} & & & \vdots \\ & & & & -\alpha_1 \end{array}\right], \quad M_2 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} \tag{11}$$

and

$$R_2 = \begin{bmatrix} g\gamma_3 & g\gamma_4 & \dots & g\gamma_{n-1} & g\gamma_{n-1} - \alpha_n \\ 0 & 0 & \dots & 0 & -\alpha_{n-1} \end{bmatrix}, \quad S_2 = \begin{bmatrix} 0 & g\gamma_2 \\ 1 & 0 \end{bmatrix}. \tag{12}$$

Applying the block triangularization procedure in [3], the following lemma is obtained.

Lemma: Let $\mu \equiv g^{-1}$. If the loop gain g satisfies

$$0 < \mu^{1/2} \leq \frac{|\gamma_2|^{1/2}}{3} (\|P_2\| + 2\|M_2\| \|S_2^{-1} R_2\|)^{-1} \tag{13}$$

then the system (9), (10) is equivalent to the block triangular system

$$\dot{z}_2 = [P_2 - M_2S_2^{-1}R_2 - M_2G_2]z_2 + M_2\eta_2 \tag{14}$$

$$\dot{\eta}_2 = [S_2 + L_2M_2]\eta_2 \tag{15}$$

where

$$\eta_2 = w_2 + L_2z_2 \tag{16}$$

$$L_2 = S_2^{-1}R_2 + G_2 \tag{17}$$

and the matrix G_2 has the representation

$$G_2 = \sum_{i=0}^{k} G_2^i\mu^{i/2} + 0(\mu^{k/2}). \tag{18}$$

Moreover, the eigenvalues of system (9), (10) are

$$\lambda(P_2 - M_2S_2^{-1}R_2) + 0(\mu) \tag{19}$$

and

$$\lambda(S_2) + 0(1). \tag{20}$$

Proof: The proof is given in the Appendix.

It is evident from the proof that $\lambda(S_2)$ is $0(g^{1/2})$, therefore, when $\gamma_2 \neq 0$, the system (8) has n-2 eigenvalues which are 0(1) and two $0(g^{1/2})$ large eigenvalues. Moreover, as $g \to \infty$, the finite eigenvalues of (8) tend to $\lambda(P_2 - M_2S_2^{-1}R_2)$ which are the n-2 zeros of the transfer function of the open loop system (1), (2). The directions of $\lambda(S_2)$ are the root locus asymptotes. The solution of (15) is given by

$$\eta_2(t) = e^{(S_2+L_2M_2)t}\eta_2(0). \tag{21}$$

Hence, the motion $\eta_2(t)$ is characterized by the $0(g^{1/2})$ large eigenvalues of (8). If $\gamma_2 < 0$, $\eta_2(t)$ has components which are exponentially growing in the $g^{1/2}t$ time scale and (8) is unstable. If $\gamma_2 > 0$, $\eta_2(t)$ has components $e^{\alpha t}(k_1\sin\sqrt{g\gamma_2}t + k_2\cos\sqrt{g\gamma_2}t)$ where α depends on the matrix L_2M_2. Clearly if $\alpha > 0$, (8) is unstable. Substitution of (16) into (4) yields

$$x = [b\ Ab]\eta_2 + N_2Z_2 \tag{22}$$

where matrix N_2 depends on Q and L_2. This shows that $\eta_2(t)$ occurs in a two dimensional subspace spanned by the vectors b and Ab.

The behavior of the high gain feedback system (8) is considered in the following theorem which includes the Lemma as a special case.

Theorem: Let $\gamma_1 = \gamma_2 = \ldots = \gamma_{q-1} = 0$ and $\gamma_q \neq 0$. $1 \leq q \leq n$. Define $(z_q)^T = (x'_{q+1}, x'_{q+2}, \ldots, x'_n) \in R^{n-q}$, $(w_q)^T = (x'_1, x'_2, \ldots, x'_q) \in R^q$,

$$P_q = \left[\begin{array}{ccc|c} 0 & 0 \ldots & 0 & -\alpha_{n-q} \\ \hline & & & -\alpha_{n-q-1} \\ & I_{n-q} & & \vdots \\ & & & -\alpha_1 \end{array}\right], \quad M_q = \begin{bmatrix} 0 & 0 & \ldots & 0 & 1 \\ 0 & 0 & \ldots & 0 & 0 \\ & & \vdots & & \\ 0 & 0 & \ldots & 0 & 0 \end{bmatrix} \tag{23}$$

$$R_q = \begin{bmatrix} g\gamma_{q+1} & g\gamma_{q+2} & \cdots & g\gamma_{n-1} & g\gamma_n - \alpha_n \\ 0 & 0 & \cdots & 0 & -\alpha_{n-1} \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & -\alpha_{q+1} \end{bmatrix} \tag{24}$$

and

$$S_q = \left[\begin{array}{cccc|c} 0 & 0 & \cdots & 0 & g\gamma_q \\ \hline & & & & 0 \\ & I_{q-1} & & & \vdots \\ & & & & 0 \end{array}\right]. \tag{25}$$

We put system (8) in the form

$$\dot{z}_q = P_q z_q + M_q w_q \tag{26}$$

$$\dot{w}_q = R_q z_q + S_q w_q \,. \tag{27}$$

If the loop gain g satisfies

$$0 < \mu^{1/q} \leq \frac{|\gamma_q|^{1/2}}{3}(\|P_q\| + 2\|M_q\|\|S_q^{-1}R_q\|)^{-1}, \tag{28}$$

then the system (26), (27) is equivalent to the block triangular system

$$\dot{z}_q = [P_q - M_q S_q^{-1} R_q - M_q G_q] z_q + M_q \eta_q \tag{29}$$

$$\dot{\eta}_q = [S_q + L_q M_q]\eta_q \tag{30}$$

where

$$\eta_q = w_q + L_q z_q \tag{31}$$

$$L_q = S_q^{-1} R_q + G_q \tag{32}$$

and the matrix G_q has the representation

$$G_q = \sum_{i=0}^{k} G_q^i \mu^{i/q} + O(\mu^{i/q}). \tag{33}$$

Moreover, the eigenvalues of (27), (28) are

$$\lambda(P_q - M_q S_q^{-1} R_q) + O(\mu) \tag{34}$$

and

$$\lambda(S_q) + O(1). \tag{35}$$

Proof: We note that the eigenvalues of S_q are distributed on a circle of radius $|g\gamma_q|^{1/q}$ in the complex plane, that is,

$$\lambda(S_q) = |g\gamma_q|^{1/q}(\cos \frac{r\pi}{q} + j \sin \frac{r\pi}{q}) \tag{36}$$

where $r = 2k\pi$ if $\gamma_q > 0$, $r = (2k+1)\pi$ if $\gamma_q < 0$, and $k = 0,1,\ldots,q-1$. Hence we can proceed analogously as in the case when $q = 2$. Following the steps in the Appendix, the block triangular form (29), (30) is obtained.

From (36), we see that $\lambda(S_q)$ is $0(g^{1/q})$. Hence, when $\gamma_1 = \ldots = \gamma_{q-1} = 0$ and $\gamma_q \neq 0$, (8) has n-q eigenvalues which are $0(1)$ and q $0(g^{1/q})$ eigenvalues. Furthermore, if $q \geq 3$, there is at least one large eigenvalue lying in the right half plane which agrees with results of Meerov [5,6]. From (34), we see that as $g \to \infty$, the finite eigenvalues of (8) tend to $\lambda(P_q - M_q S_q^{-1} R_q)$ which are the n-q zeros of the transfer function of the open loop system (1), (2). The root locus asymptotes are given by the directions of $\lambda(S_q)$ which are straight lines forming angle $\frac{r\pi}{q}$ with the real axis. Finally, the solution of (30) is given by

$$\eta_q(t) = e^{(S_q+L_q M_q)t}\, \eta_q(0). \tag{37}$$

Analogous to (22),

$$x = [b, Ab, \ldots, A^{q-1}b]\eta_q + N_q z_q \tag{38}$$

where N_q depends on Q and L_q. This illustrates that $\eta_q(t)$ occurs in a q-dimensional subspace spanned by the first q vectors of the controllability matrix Q.

The theorem also shows that the high gain feedback system (8) can be classified by means of the integer q. Given the matrix $c^T Q$, we then know the values of γ_i since by definition $\gamma_i = c^T A^{i-1} b$. Hence the following classification is appropriate.

Definition: The single input-single output system (1), (2) is said to belong to class q if the system matrices satisfy the relationships

$$c^T A^{i-1} b = 0,\ i = 1, \ldots, q-1,\quad c^T A^{q-1} b \neq 0. \tag{39}$$

A class q system then has the following properties:

(i) There are n-q zeros in its transfer function.

If output feedback (3) is applied, then

(ii) the root locus asymptotes are straight lines making angles $\frac{r\pi}{q}$,

$$r = \begin{cases} 2k\pi & \text{if } c^T A^{q-1} b > 0 \\ (2k+1)\pi & \text{if } c^T A^{q-1} b < 0 \end{cases},\quad k = 0, 1, \ldots, q \tag{40}$$

with the real axis,

(iii) there are q $0(g^{1/q})$ large eigenvalues among the closed loop eigenvalues.

From the definition, it is seen that the classification procedure involves the computation of at most n-1 matrix products and n inner products.

Example: As an illustration, consider the system

$$\dot{x} = \begin{bmatrix} .2 & .2 & 0 & .8 \\ 1 & 0 & .4 & 2.8 \\ 3.6 & .6 & 3.2 & -.2 \\ 1.2 & .2 & 1.4 & .6 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \\ 3 \\ 1 \end{bmatrix} u \tag{41}$$

$$y = [2.6 \quad -.4 \quad .8 \quad -2]x. \tag{42}$$

By applying the transformation Q (5), we found that

$$[-\alpha_4, -\alpha_3, -\alpha_2, -\alpha_1] = [1,2,-1,4] \tag{43}$$

$$[c^Tb \;\; c^TAb \;\; c^TA^2b \;\; c^TA^3b] = [0,-1,-3,-8]. \tag{44}$$

Since $c^Tb = 0$ and $c^TAb < 0$, we conclude that the high gain feedback system, (41), (42) with $u = gy$ belongs to class 2. Hence there are two zeros in the transfer function $\frac{y(s)}{u(s)}$. Moreover, the large eigenvalues are $O(g^{1/2})$ and the root locus asymptotes are straight lines making angles $\frac{\pi}{2}$ and $\frac{3\pi}{2}$ with the real axis. The closed loop eigenvalues are computed by taking first order approximations of (35), (36), that is,

$$\lambda_{1,2} \cong \lambda(P_2 - M_2S_2^{-1}R_2 - M_2G_2^o) = \lambda\left(\begin{bmatrix} -3 & -9 \\ 1 & 4 \end{bmatrix} - \begin{bmatrix} 2\mu & 8\mu \\ 0 & 0 \end{bmatrix}\right) \tag{45}$$

$$= \frac{1}{2} + \mu \pm \frac{1}{2}\sqrt{13 + 4\mu + 4\mu^2}$$

and

$$\lambda_{3,4} \cong \lambda(S_2 + S_2^{-1}R_2M_2) = \lambda\left[\begin{pmatrix} 0 & -g \\ 1 & 0 \end{pmatrix} + \begin{pmatrix} 3 & 0 \\ 0 & 0 \end{pmatrix}\right] = \frac{3}{2} \pm \frac{1}{2}\, j\sqrt{4g-9} \tag{46}$$

The exact eigenvalues are computed for comparison purposes. Denoting the error between the approximate and exact eigenvalues as ε, the results are summarized in the following table

Gain g	Exact	Approximate	ε%
20	2.6110325	2.3672782	9.34
	-1.1510292	-1.2672782	10.10
	1.2699983	1.5	
	±j4.3225080	±j4.2130749	.74
100	2.3620711	2.3065747	2.4
	-1.2641928	-1.3045747	3.2
	1.4510608	1.5	
	±j9.9345038	±j9.8868600	.40
1000	2.3086180	2.3031532	.24
	-1.2986416	-1.3029532	.34
	1.4950118	1.5	
	±j31.602899	±j31.587181	.05

The lower bound on $\sqrt{g}$ given in (13) is 66. We see that the approximations on the eigenvalues given by (45), (46) are reasonably good even for $g = 20$. As $g \to \infty$, the finite eigenvalues $\lambda_{1,2}$ tend to $\frac{1}{2}(1 \pm \sqrt{13})$ which can be verified to be the zeros of system (41), (42).

3. CONCLUSIONS

We have shown that by transforming the open loop system (1), (2) into the canonical from (6), (7), singular perturbation analysis can be readily applied to investigate the asymptotic behavior of the high gain feedback system (8) when the loop gain is large. It is shown that the closed loop eigenvalues consist of two groups, a group which is of 0(1) magnitudes and depending on the system matrices, the magnitudes of the second group increase with different speeds as the loop gain increases. The computation of these eigenvalues is decomposed into two separate lower order problems. Finally, a classification of single input-single output systems which based on their structures is proposed.

We note that the singular perturbation approach presented in this paper for the analysis of single input-single output high gain systems can be extended to the multivariable case. Some results have been obtained in this respect and will be reported in a later paper.

APPENDIX

We first note that the eigenvalues of the matrix S_2 are not zero, hence there exists a transformation J_2,

$$w_2' = J_2^{-1} w_2 \tag{A1}$$

such that the system (9), (10) is equivalent to the system

$$\dot{z}_2 = P_2 z_2 + M_2' w_2' \tag{A2}$$

$$\mu^{1/2} \dot{w}_2' = R_2' z_2 + S_2' w_2' \tag{A3}$$

where

$$S_2' = J_2^{-1} S_2 J_2 = \begin{cases} \begin{bmatrix} 0 & \gamma_2^{1/2} \\ -\gamma_2^{1/2} & 0 \end{bmatrix} & \text{if } \gamma_2 < 0 \\ \begin{bmatrix} \gamma_2^{1/2} & 0 \\ 0 & -\gamma_2^{1/2} \end{bmatrix} & \text{if } \gamma_2 > 0 \end{cases} \tag{A4}$$

and

$$R_2' = \mu^{1/2}(J_2^{-1} R_2), \quad M_2' = M_2 J_2. \tag{A5}$$

Since S_2' is nonsingular, the system (A2), (A3) is in the standard singularly perturbed form. By identifying $\mu^{1/2}$ as the parasitic parameter, the block triangularization procedures in [3,4] can be applied and results in the block triangular system given by

$$\dot{z}_2 = [P_2 - M_2 S_2^{-1} R_2 - \mu^{1/2} M_2' G_2'] z_2 + M_2' \eta_2' \tag{A6}$$

$$\mu^{1/2} \dot{\eta}_2' = [S_2' + \mu^{1/2} L_2' M_2'] \eta_2' \tag{A7}$$

with

$$\eta_2' = w_2' + L_2 z_2 \tag{A8}$$

$$L_2' = S_2'^{-1} R_2' + \mu^{1/2} G_2' \tag{A9}$$

$$G_2' = \sum_{i=0}^{k} G_2'^{\,i} \mu^{i/2} + 0(\mu^{k/2}) \qquad \text{(A10)}$$

and the matrix coefficients $G_2'^{\,i}$ are obtained from a recursive formula

$$G_2'^{\,o} = S_2'^{\,-2} R_2' [P_2 - M_2 S_2^{-1} R_2] \qquad \text{(A11)}$$

$$G_2'^{\,k+1} = S_2'^{\,-1} [G_2'^{\,k+1} (P_2 - M_2 S_2^{-1} R_2) - S_2'^{\,-1} R_2' M_2' G_2'^{\,k+1} - \sum_{i=0}^{k} G_2'^{\,i} M_2' G_2'^{\,k-i}], \qquad \text{(A12)}$$
$$k \geq -1, \quad \sum_{i=0}^{-1} \equiv 0.$$

By (A4), (A5), (A9), (A10)

$$g^{1/2} S_2' + L_2' M_2' = J_2^{-1} (S_2 + L_2 M_2) J_2 \qquad \text{(A13)}$$

and

$$M_2' G_2' = g^{1/2} M_2 G_2, \quad G_2' = g^{1/2} J_2^{-1} G_2. \qquad \text{(A14)}$$

By defining

$$\eta_2 = J_2 \eta_2' \qquad \text{(A15)}$$

(A6), (A7) becomes (14), (15). Finally from Lemma 1 of [3] the estimate on $\mu^{1/2}$ is given by

$$0 < \mu^{1/2} \gamma_2^{1/2} \leq \frac{1}{3} (\|P_2\| + 2\|M_2'\| \|S_2'^{\,-1} R_2'\|)^{-1} \qquad \text{(A16)}$$

which induces (13). The approximations (19), (20) of the eigenvalues are obtained by observing that $\|M_2 G_2\| = 0(g^{-1})$ and $\|L_2 M_2\| = 0(1)$.

ACKNOWLEDGMENT

The author wishes to thank Professor S. Bingulac, whose suggestion to use transformation Q of (5) is instrumental to the development of the results presented.

REFERENCES

1. W. R. Evans, "Graphical Analysis of Control Systems," Trans. AIEE, Vol. 67, 1948, pp. 547-551.
2. K.-K. D. Young, P. V. Kokotovic and V. I. Utkin, "A Singular Perturbation Analysis of High Gain Feedback Systems," Proc. 1977 Joint Automatic Control Conference, pp. 1270-1277.
3. P. V. Kokotovic, "A Riccati Equation for Block-Diagonalization of Ill-Conditioned Systems," IEEE Trans. on Automatic Control, Vol. AC-20, No. 6, December 1975, pp. 812-814.
4. P. V. Kokotovic and A. H. Haddad, "Controllability and Time Optimal Control of Systems with Slow and Fast Modes," IEEE Trans. on Automatic Control, Vol. AC-20, No. 1, February 1975, pp. 111-113.

5. M. V. Meerov, "Automatic Control Systems Stable with Indefinitely Large Values of Gain," Automatika i Telemekhanika, Vol. 8, No. 4, 1947.

6. M. V. Meerov, Structural Synthesis of High Accuracy Automatic Control Systems, Pergamon Press, 1965.

STATE VARIABLE RECONSTRUCTION OF LINEAR DISCRETE SYSTEMS FROM DELAYED OBSERVATIONS

DONG H. CHYUNG
Electrical and Computer Engineering
University of Iowa
Iowa City, Iowa 52242

ABSTRACT

The problem of reconstructing the present state variable from delayed partial measurements of the state variable and the input function is studied.

For the sake of simplicity consider the linear discrete system

$$x(k + 1) = Ax(k) + Bu(k) \quad , \qquad x(0) = x_o \tag{1}$$

where $x \varepsilon R^n$ is the state variable, $u \varepsilon R^m$ is the control variable, A and B are constant matrices of compatible dimensions, and k is a positive integer.

Suppose the only quantities which are observable at k are

$$y_i(k) = H_i x(k - h_i) \quad , \qquad i = 0, 1, 2, \ldots, N \quad ,$$

where, for each i, $y_i(k)$ is an m_i-dimensional vecotr, H_i is an $m_i \times n$ constant matrix, and $0 = h_o < h_1 < \ldots < h_N$ are integer delays.

The problem is to find $x(k)$ from the observed variables, $y_o(k)$, $y_1(k), \ldots, y_N(k)$ and $u(j)$, $j = k - h_N, k - h_N + 1, \ldots, k - 1$.

Now, for $i = 0, 1, \ldots, N$,

$$x(k) = A^{h_i} x(k - h_i) + \sum_{i=1}^{h_i} A^{j-1} Bu(k-j).$$

If A^{h_i} is nonsingular,

$$\left(A^{h_i}\right)^{-1} x(k) = x(k - h_i) + \left(A^{h_i}\right)^{-1} \sum_{j=1}^{h_i} A^{j-1} Bu(k - j)$$ and so, for

$i = 0, 1, \ldots, N,$

$$H_i\left(A^{h_i}\right)^{-1} x(k) = H_i x(k - h_i) + H_i\left(A^{h_i}\right)^{-1} \sum_{j=1}^{h_i} A^{j-1} Bu(k - j) \quad . \tag{2}$$

Since the right hand side is known, $x(k)$ can be obtained from $y_i(k)$, $i = 0, 1, 2, \ldots, N$, and $u(j)$, $j = k - h_N, \ldots, k - 1$, if and only if the matrix

$$\begin{bmatrix} H_o \\ H_i\left(A^{h_i}\right)^{-1} \\ \cdot \\ \cdot \\ \cdot \\ H_N\left(A^{h_N}\right)^{-1} \end{bmatrix}$$

has rank n, and $x(k)$ is obtained by solving the simultaneous equations (2).

ZERO-SUM NASH STRATEGY FOR SYSTEMS WITH FAST AND SLOW MODES

B. F. GARDNER, JR.
Decision and Control Laboratory
Coordinated Science Laboratory
University of Illinois, Urbana, Illinois

ABSTRACT

For singularly perturbed systems with zero-sum cost functionals, a decomposition into two lower order zero-sum games is formulated. A complete separation of fast and slow subsystems is achieved which leads to a first order approximation of the optimal feedback controls and a second order approximation of the optimal cost without knowledge of the small singular perturbation parameter.

1. INTRODUCTION

For control problems of high dimension, a major difficulty in computing the optimal control is that the dimension of the nonlinear Riccati equation is large. In game problems this "curse of dimensionality" is even more pronounced. Attempts have been made to reduce the order of the Riccati equations to be solved both in control [1-4] and in dynamic games [5]. In optimal control, recent results [6] demonstrate that for singularly perturbed systems with fast and slow modes the system can be decomposed into two systems of lower order and, with a savings in computation, a near optimal control can be derived which gives a good approximation to the optimal cost when applied. In dynamic games, two methods have been used. In [5] a parameter imbedding technique is used with series expansions and truncation to give approximations to the Riccati gains. In [7], for a special system representation, singular perturbation techniques have been applied which give Pareto optimal and Nash equilibrium strategies which result in first order approximations to the optimal cost functions.

This paper considers the zero-sum Nash game for a singularly perturbed system based on the results presented in [6]. By a decomposition into two lower order zero-sum games corresponding to the fast and slow modes of the original system, a composite control will be derived which is an $0(\mu)$ approximation of the Nash optimal control and leads to an $0(\mu^2)$ approximation to the optimal cost.

2. LINEAR QUADRATIC ZERO-SUM NASH SINGULARLY PERTURBED GAME

2.1. Problem Statement

Consider a singularly perturbed time invariant system (1)-(2)

$$\dot{x}_1 = A_{11}x_1 + A_{12}x_2 + B_{11}u_1 + B_{12}u_2, \qquad x_1(0) = x_{10} \tag{1}$$

$$\mu\dot{x}_2 = A_{21}x_1 + A_{22}x_2 + B_{21}u_1 + B_{22}u_2, \qquad x_2(0) = x_{20} \tag{2}$$

where x_1, x_2, u_1, and u_2 are n_1, n_2, m_1, and m_2 vectors respectively and μ is a small positive scalar. Players one and two control u_1 and u_2 respectively which they choose based on a zero-sum linear quadratic Nash criterion given by

$$J_1 = -J_2 = \frac{1}{2}\int_{t_0}^{t_f}\left\{ \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}' \begin{bmatrix} Q_1 & Q_2 \\ Q_2' & Q_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + u_1'R_1u_1 + u_2'R_2u_2 \right\} dt. \tag{3}$$

Player one desires to minimize J_1 while player two desires to minimize J_2. Since $J_1 + J_2 = 0$ players one and two are in direct conflict and any improvement in the value of J_1 for player one constitutes a worsening of the value of J_2 for player two.

The Riccati equations and controls for system (1)-(2) subject to (3) are well known [9] and are given by (4)-(5).

$$\dot{S} = -A'S - SA - Q + S[B_1R_1^{-1}B_1' + B_2R_2^{-1}B_2']S, \qquad S(t_f) = 0 \tag{4}$$

where

$$A = \begin{bmatrix} A_{11} & A_{12} \\ \frac{A_{21}}{\mu} & \frac{A_{22}}{\mu} \end{bmatrix}, \quad B_1 = \begin{bmatrix} B_{11} \\ \frac{B_{21}}{\mu} \end{bmatrix}, \quad B_2 = \begin{bmatrix} B_{12} \\ \frac{B_{22}}{\mu} \end{bmatrix}, \text{ and } Q = \begin{bmatrix} Q_1 & Q_2 \\ Q_2' & Q_3 \end{bmatrix}$$

$$u_i = -R_i^{-1}B_i'S \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \qquad i = 1,2. \tag{5}$$

To avoid unboundedness in (4) as $\mu \to 0$ the solution is sought in the form

$$S = S(\mu) = \begin{bmatrix} S_1(\mu) & \mu S_2(\mu) \\ \mu S_2'(\mu) & \mu S_3(\mu) \end{bmatrix} \tag{6}$$

which permits setting $\mu = 0$ in (4). Setting $\mu = 0$ yields a Riccati equation and two algebraic equations given by (7), (8) and (9).

$$\begin{aligned} \dot{S}_1 = {} & -Q_1 - A_{11}'S_1 - S_1A_{11} - A_{21}'S_2' - S_2A_{21} + S_1(B_{11}R_1^{-1}B_{11}' + B_{12}R_2^{-1}B_{12}')S_1 \\ & + S_1(B_{11}R_1^{-1}B_{21}' + B_{12}R_2^{-1}B_{22}')S_2' + S_2(B_{21}R_1^{-1}B_{11}' + B_{22}R_2^{-1}B_{12}')S_1 \\ & + S_2(B_{21}R_1^{-1}B_{21}' + B_{22}R_2^{-1}B_{22}')S_2' \end{aligned} \tag{7}$$

$$\begin{aligned} 0 = {} & -Q_2 - A_{21}'S_3 - S_1A_{12} - S_2A_{22} + (S_1B_{11} + S_2B_{21})R_1^{-1}B_{21}'S_3 \\ & + (S_1B_{12} + S_2B_{22})R_2^{-1}B_{22}'S_3 \end{aligned} \tag{8}$$

$$0 = -Q_3 - S_3A_{22} - A_{22}'S_3 + S_3(B_{21}R_1^{-1}B_{21}' + B_{22}R_2^{-1}B_{22}')S_3. \tag{9}$$

Solving equations (7), (8) and (9) is clearly a difficult undertaking in general. When $\mu \neq 0$ in the system (1)-(2) and in the Riccati equations (4) and controls, solving the corresponding problem is clearly even more difficult since there would then be three sets of coupled Riccati equations. Furthermore, in the nonzero-sum case the number of coupled Riccati equations is doubled to six so that the "curse of dimensionality" is readily apparent. In this section it is desired to take advantage of the two-time scale property of the system (1)-(2) where, analagous to the control problem [6], there are two sets of eigenvalues of order $0(1)$ and $0(1/\mu)$. By reducing the order of (1)-(2) a "composite" control is designed which is an $0(\mu)$ approximation to the full order control. When this composite control is implemented it leads to an $0(\mu^2)$ approximation of the optimal cost.

2.2. Slow and Fast Subsystems

Neglecting the fast modes in (1)-(2) by setting $\mu = 0$ gives a reduced order system involving only the slow modes. Defining $x_s = \bar{x}_1$, $u_{1s} = \bar{u}_1$, $u_{2s} = \bar{u}_2$ and $\bar{x}_2$ as the slow parts of the corresponding variables, where the bar indicates that $\mu = 0$, the "slow" subsystem of (1)-(2), assuming A_{22} is nonsingular, is

$$\dot{x}_s = A_o x_s + B_{01} u_{1s} + B_{02} u_{2s}, \qquad x_s(0) = x_{10} \tag{10}$$

where

$$A_o = A_{11} - A_{12} A_{22}^{-1} A_{21}$$

$$B_{0i} = B_{1i} - A_{12} A_{22}^{-1} B_{2i}, \qquad i = 1,2.$$

To derive the "fast" part of (1)-(2) corresponding to the large eigenvalues, let the slow variables be considered constant during the fast transient. Defining $x_f = x_2 - \bar{x}_2$, $u_{1f} = u_1 - u_{1s}$ and $u_{2f} = u_2 - u_{2s}$ the fast part of (1)-(2) is

$$\mu \dot{x}_f = A_{22} x_f + B_{21} u_{1f} + B_{22} u_{2f}, \qquad x_f(0) = x_{20} - \bar{x}_2(0). \tag{11}$$

If $u_{is} = G_{0i} x_s$ and $u_{if} = G_{2i} x_f$, $i = 1,2$, where G_{0i} and G_{2i} are some known gains, then the composite control $u_{ic} = u_{is} + u_{if}$ can be written as

$$\begin{aligned} u_{ic} = {} & ((I + G_{2i} A_{22}^{-1} B_{2i}) G_{0i} + G_{2i} A_{22}^{-1} A_{21} + G_{2i} A_{22}^{-1} B_{2j} G_{0j}) x_s \\ & + G_{2i}(-A_{22}^{-1}(A_{21} + B_{2i} G_{0i} + B_{2j} G_{0j}) x_s + x_f), \quad i,j = 1,2, \quad j \neq i. \end{aligned} \tag{12}$$

Lemma 1: If the controls

$$u_{is} = G_{0i} x_s \quad , \qquad i,j = 1,2, \quad i \neq j \tag{13}$$

$$u_{if} = G_{2i}x_f \quad , \quad i,j = 1,2, \quad i \neq j \tag{14}$$

$$\begin{aligned} u_i &= ((I + G_{2i}A_{22}^{-1}B_{2i})G_{0i} + G_{2i}A_{22}^{-1}A_{21} + G_{2i}A_{22}^{-1}B_{2j}G_{0j})x_1 \\ &\quad + G_{2i}x_2 \quad , \quad i,j = 1,2, \quad i \neq j \end{aligned} \tag{15}$$

are applied to systems (10), (11) and (1)-(2) respectively and if $(A_{22} + B_{21}G_{21} + B_{22}G_{22})$ is stable then

$$x_1(t) = x_s(t) + 0(\mu) \tag{16}$$

$$x_2(t) = -A_{22}^{-1}(A_{21} + B_{21}G_{01} + B_{22}G_{02})x_s(t) + x_f(t) + 0(\mu) \tag{17}$$

$$u_i(t) = u_{is}(t) + u_{if}(t) + 0(\mu), \qquad i = 1,2. \tag{18}$$

Proof: Apply (15) to the system (1)-(2) and diagonalize using the method of [11]. The time functions of the resulting states are then compared with the time functions of the states resulting from application of (13) to (10) and (14) to (11). The results given by equations (16), (17) and (18) follow immediately.

As in the control case, Lemma 1 suggests that G_{0i} and G_{2i}, $i = 1,2$ can be designed separately and then used in a composite control defined by (15). G_{0i}, $i = 1,2$, can be designed based on some desired "slow" performance criterion and G_{2i}, $i = 1,2$, can be designed based on some desired "fast" performance criterion.

2.3. Well-Posedness of the Reduced Game

For the slow subsystem given by (10) consider a "reduced" cost (19) involving only slow variables.

$$\begin{aligned} J_{1s} = -J_{2s} &= \frac{1}{2}\int_{t_0}^{t_f} (x_s'\hat{Q}_1x_s + 2x_s'\hat{Q}_2(B_{21}u_{1s} + B_{22}u_{2s}) + u_{1s}'\hat{R}_1u_{1s} \\ &\quad + u_{2s}'\hat{R}_2u_{2s} + 2u_{1s}'\hat{Q}_3u_{2s})dt \end{aligned} \tag{19}$$

where

$$\begin{aligned} \hat{Q}_1 &= Q_1 - Q_2A_{22}^{-1}A_{21} - (A_{22}^{-1}A_{21})'Q_2' + (A_{22}^{-1}A_{21})'Q_3A_{22}^{-1}A_{21} \\ \hat{Q}_2 &= (A_{22}^{-1}A_{21})'Q_3A_{22}^{-1} - Q_2A_{22}^{-1} \\ \hat{R}_1 &= R_1 + B_{21}'(A_{22}^{-1})'Q_3A_{22}^{-1}B_{21} \\ \hat{R}_2 &= R_2 + B_{22}'A_{22}'^{-1}Q_3A_{22}^{-1}B_{22} \\ \hat{Q}_3 &= B_{21}'(A_{22}^{-1})'Q_3A_{22}^{-1}B_{22}. \end{aligned} \tag{20}$$

The Nash controls for the game (10), (19) are easily found to be

$$\begin{bmatrix} u_{1s} \\ u_{2s} \end{bmatrix} = - \begin{bmatrix} \hat{R}_1 & \hat{Q}_3 \\ \hat{Q}'_3 & \hat{R}_2 \end{bmatrix}^{-1} \begin{bmatrix} B'_{21}\hat{Q}'_2 + B'_{01}K_s \\ B'_{22}\hat{Q}'_2 + B'_{02}K_s \end{bmatrix} \tag{21}$$

where

$$\dot{K}_s = -\hat{Q}_1 - K_sA_o - A'_oK_s + \begin{bmatrix} B'_{22}\hat{Q}'_2 + B'_{02}K_s \\ B'_{21}\hat{Q}'_2 + B'_{01}K_s \end{bmatrix}' \begin{bmatrix} \hat{R}_2 & \hat{Q}'_3 \\ \hat{Q}_3 & \hat{R}_1 \end{bmatrix}^{-1} \begin{bmatrix} B'_{22}\hat{Q}'_2 + B'_{02}K_s \\ B'_{21}\hat{Q}'_2 + B'_{01}K_s \end{bmatrix} \tag{22}$$

Theorem 1: If A_{22}, R_1, R_2 and

$$\begin{bmatrix} R_2 - B'_{22}S_3A_{22}^{-1}B_{22} & -B'_{22}S_3A_{22}^{-1}B_{21} \\ -B'_{21}S_3A_{22}^{-1}B_{22} & R_1 - B'_{21}S_3A_{22}^{-1}B_{21} \end{bmatrix}$$

are invertible for $t \in [t_o, t_f]$ then

$$K_s = S_1 \tag{23}$$

where S_1 is the solution to equation (7).

Proof: Elimination of S_3 and S_2 from equation (7) through manipulation of equations (7), (8) and (9) gives an equation in S_1 which is the same as equation (22).

Theorem 1 demonstrates that the reduced zero-sum game is well-posed. That is, as $\mu \to 0$ in two different ways the same Riccati gain results. In order to find a reduced order Riccati gain at $\mu = 0$ it is not necessary to consider the set of one Riccati and two algebraic equations (7)-(9) but rather one can consider a reduced system and a reduced cost functional corresponding to the slow modes of the full order system. Thus an $O(\mu)$ approximation of part of the full order Riccati gain can be found with a savings in computation.

2.4. Near Optimality of the Composite Control and Corresponding Cost

Consider the fast subsystem of (1)-(2) given by equation (11). To design the controls u_{1f} and u_{2f} a criterion is needed. Consider the "fast" cost

$$J_{1f} = -J_{2f} = \frac{1}{2}\int_{t_0}^{\infty} (x'_fQ_3x_f + u'_{1f}R_1u_{1f} + u'_{2f}R_2u_{2f})dt \tag{24}$$

which is obtained by substituting $x_2 = x_f + x_{2s}$, $u_1 = u_{1f} + u_{1s}$ and $u_2 = u_{2f} + u_{2s}$ into (3) and retaining only the quadratic fast terms. The Nash controls for (11) subject to (25) are

$$u_{if} = -R_i^{-1} B'_{2i} K_f x_f, \qquad i = 1,2 \tag{25}$$

where

$$0 = -Q_3 - K_f A_{22} - A'_{22} K_f + K_f (B_{21} R_1^{-1} B'_{21} + B_{22} R_2^{-1} B'_{22}) K_f. \tag{26}$$

Comparison of (26) and (9) shows that $K_f = S_3$.

Choosing G_{0i}, $i = 1,2$, from equation (21) and G_{2i}, $i = 1,2$, from equation (25) the composite control is given by

$$\begin{aligned}\begin{bmatrix} u_{1c} \\ u_{2c} \end{bmatrix} = &- \begin{bmatrix} R_1^{-1} & 0 \\ 0 & R_2^{-1} \end{bmatrix} \begin{bmatrix} R_1 - B'_{21} K_f A_{22}^{-1} B_{21} & -B'_{21} K_f A_{22}^{-1} B_{22} \\ -B'_{22} K_f A_{22}^{-1} B_{21} & R_2 - B'_{22} K_f A_{22}^{-1} B_{22} \end{bmatrix} \\ &\cdot \begin{bmatrix} \hat{R}_1 & \hat{Q}_3 \\ \hat{Q}'_3 & \hat{R}_2 \end{bmatrix}^{-1} \begin{bmatrix} B'_{21} \hat{Q}'_2 + B'_{01} K_s \\ B'_{22} \hat{Q}'_2 + B'_{02} K_s \end{bmatrix} x_1 - \begin{bmatrix} R_1^{-1} B'_{21} K_f A_{22}^{-1} A_{21} \\ R_2^{-1} B'_{22} K_f A_{22}^{-1} A_{21} \end{bmatrix} x_1 \\ &- \begin{bmatrix} R_1^{-1} B'_{21} K_f \\ R_2^{-1} B'_{22} K_f \end{bmatrix} x_2. \end{aligned} \tag{27}$$

The full order control with $\mu = 0$ is given by

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = - \begin{bmatrix} R_1^{-1} & 0 \\ 0 & R_2^{-1} \end{bmatrix} \begin{bmatrix} B'_{11} S_1 + B'_{21} S'_2 \\ B'_{12} S_1 + B'_{22} S'_2 \end{bmatrix} x_1 - \begin{bmatrix} R_1^{-1} B'_{21} S_3 \\ R_2^{-1} B'_{22} S_3 \end{bmatrix} x_2. \tag{28}$$

Theorem 2: Under the conditions of Theorem 1

$$u_{ic}(t) = u_i(t) + O(\mu). \tag{29}$$

Proof: Replacing S_3 by K_f and S_2 from equation (8) in (28) gives an equation which is identical to equation (27). The result follows easily.

Theorem 2 states that the composite control is near optimal. If this composite control is applied to (1)-(2) the cost is $J_c = \frac{1}{2} x'_0 P_c x_0$ where P_c is the positive semidefinite solution of

$$\begin{aligned} \dot{P}_c = &-Q - M'_c (B_1 R_1^{-1} B'_1 + B_2 R_2^{-1} B'_2) M_c - P_c (A - (B_1 R_1^{-1} B'_1 + B_2 R_2^{-1} B'_2) M_c) \\ &- (A' - M'_c (B_1 R_1^{-1} B'_1 + B_2 R_2^{-1} B'_2)) P \end{aligned} \tag{30}$$

$$P_c(t_f) = 0$$

where

$$M_c = \begin{bmatrix} K_s & 0 \\ \mu K_m' & \mu K_f \end{bmatrix}$$

and

$$K_m = \hat{Q}_2(A_{22} - B_{21}R_1^{-1}B_{21}'K_f - B_{22}R_2^{-1}B_{22}'K_f)A_{22}^{-1} + A_{21}'(A_{22}^{-1})'K_f$$
$$+ K_s(-A_{12} - B_{01}R_1^{-1}B_{21}'K_f - B_{02}R_2^{-1}B_{22}'K_f)A_{22}^{-1} = S_2. \qquad (31)$$

Theorem 3: If P_c and S have power series expansions about $\mu = 0$, i.e.,

$$P_c = \sum_{i=0}^{\infty} \frac{\mu^i}{i!} \begin{bmatrix} P_{c1}^{(i)} & \mu P_{c2}^{(i)} \\ \mu P_{c2}'^{(i)} & \mu P_{c3}^{(i)} \end{bmatrix}$$

$$S = \begin{bmatrix} S_1 & \mu S_2 \\ \mu S_2' & \mu S_3 \end{bmatrix} + \sum_{i=1}^{\infty} \frac{\mu^i}{i!} \begin{bmatrix} S_1^{(i)} & \mu S_2^{(i)} \\ \mu S_2'^{(i)} & \mu S_3^{(i)} \end{bmatrix}$$

then when u_{1c} and u_{2c} are applied to (1)-(2)

$$J_c = J_{opt} + O(\mu^2). \qquad (32)$$

Proof: Construct a Lyapunov equation by subtracting S from P_c (equation (30)). Manipulation then yields the result in (32).

3. CONCLUSIONS

For a singularly perturbed zero-sum dynamic Nash game a decomposition into two lower order subsystems has been demonstrated. From the Riccati equations for these lower order subsystems gains have been found which can be applied to the full order system through composite controls which are $O(\mu)$ approximations of the optimal controls. The resulting cost has been shown to be an $O(\mu^2)$ approximation of the optimal cost. Moreover, this simplification in computation is independent of μ and the derivation of the fast gain and the slow gain are mutually independent.

4. ACKNOWLEDGMENTS

The author wishes to express his gratitude to Professor J. B. Cruz, Jr. of the University of Illinois, Urbana, for his suggestions and guidance in the course of this research. This work was supported in part by the National Science Foundation under Grant ENG 74-20091, in part by the Energy Research and Development Administration, Electric Energy Systems Division under Contract EX-76-C-01-2088, and in part by the Joint Services Electronics Program (U.S. Army, U.S. Navy, and U.S. Air Force) under Contract DAAB-07-72-C-0259.

REFERENCES

[1] P. V. Kokotovic, W. R. Perkins, J. B. Cruz, Jr., and G. D'Ans, "ε-Coupling Method for Near-Optimum Design of Large Scale Linear Equations," Proceedings IEE, Vol. 116, No. 5, May 1969, pp. 889-892.

[2] P. V. Kokotovic, R. E. O'Malley, Jr., and P. Sannuti, "Singular Perturbations and Order Reduction in Control Theory - An Overview," Automatica, Vol. 12, 1976, pp. 123-132.

[3] R. R. Wilde and P. V. Kokotovic, "Optimal Open- and Closed-Loop Control of Singularly Perturbed Linear Systems," IEEE Transactions on Automatic Control, Vol. AC-18, No. 6, December 1973.

[4] R. A. Yackel and P. V. Kokotovic, "A Boundary Layer Method for the Matrix Riccati Equation," IEEE Transactions on Automatic Control, Vol. AC-18, No. 1, February 1973.

[5] J. B. Cruz, Jr. and C. I. Chen, "Series Nash Solution of Two-Person, Nonzero-Sum, Linear-Quadratic Differential Games," JOTA, Vol. 7, No. 4, April 1971, pp. 240-257.

[6] J. Chow and P. V. Kokotovic, "A Decomposition of Near-Optimum Regulators for Systems with Slow and Fast Modes," IEEE Transactions on Automatic Control, October 1976.

[7] H. K. Khalil and P. V. Kokotovic, "Control Strategies for Decision Makers Using Different Models of the Same System," submitted for publication, April 1977.

[8] J. B. Cruz, Jr., "Survey of Nash and Stackelberg Equilibrium Strategies in Dynamic Games," Ann. Econ. Soc. Meas., Vol. 4, pp. 339-344, 1975.

[9] A. W. Starr, "Nonzero-Sum Differential Games: Concepts and Models," Ph. D. Thesis, Harvard University, Technical Report No. 590, June 1969.

[10] A. H. Haddad and P. V. Kokotovic, "Note on Singular Perturbation of Linear State Regulators," IEEE Transactions on Automatic Control, June 1971.

[11] P. V. Kokotovic and A. H. Haddad, "Controllability and Time Optimal Control of Systems with Slow and Fast Modes," IEEE Transactions on Automatic Control, February 1975.

[12] I. B. Rhodes and D. G. Luenberger, "Stochastic Differential Games with Constrained State Estimators," IEEE Transactions on Automatic Control, Vol. AC-14, No. 5, October 1969.

TRANSFORMATION OF LINEAR MULTIVARIABLE SYSTEMS TO CANONICAL FORMS

J. HICKIN and N.K. SINHA
Research Group in Simulation, Optimization and Control
Faculty of Engineering, McMaster University
Hamilton, Ontario, Canada

ABSTRACT

An algorithm is presented which enables the transformation of the state equations of linear multivariable systems to any of six different canonical forms. An interesting feature of this algorithm is that no matrix inversion is required, nor is it necessary to perform extensive tests for the linear independence of sets of vectors.

1. INTRODUCTION

In a recent paper [1] six different canonical forms for linear multivariable systems were described, each of which has certain practical advantages, when applied to identification, pole-placement by state-feedback, decoupling, design of observers and model-matching. Methods for transformation of a given set of state equations to these canonical forms were also discussed. These were based primarily on the work of Luenberger [2] and require the determination of a set of linearly independent vectors from the controllability or the observability matrix of the system, selected in a particular order. The transformation matrix consisting of these vectors has then to be inverted to obtain either the column-companion, the row-companion, the input-identifiable or the output-identifiable form. The transformation to the controllable or the observable canonical form requires the inversion of two n x n matrices, where n is the order of the system.

The object of this paper is to present an efficient method for transformation to these canonical forms which does not require matrix inversion. The proposed method is based on the algorithm for the reduction of a matrix to its Hermite normal form [3, 4]. Only one such reduction is required for transformation to either the column-companion, the row-companion, the output-identifiable or the input-identifiable form. The transformation to the controllable or the observable canonical form requires two such reductions. In all these cases, the procedure is computationally superior to other methods [5, 6] which have been proposed recently.

2. TRANSFORMATION TO THE COLUMN-COMPANION FORM

Consider a linear multivariable system described by the equations

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{1}$$

where $x \in R^n$, $u \in R^p$ and $y \in R^q$. It will be assumed that this is a minimal realization.

The proposed method starts by first obtaining the matrix S, given by

$$\begin{aligned} S &= [b_1 \; Ab_1 \; \ldots \; A^n b_1 \; b_2 \; Ab_2 \; \ldots \; A^{n-1} b_2 \; \ldots \; b_p \; Ab_p \; \ldots \; A^{n-p+1} \; b_p] \\ &= [s_1 \; s_2 \; \ldots \; s_{n+1} \; s_{n+2} \; \ldots \; s_{np+n-\frac{1}{2}p^2+\frac{1}{2}p}] \end{aligned} \tag{3}$$

where b_i is the ith column of B.

The matrix S is now transformed to the Hermite normal form using the following procedure. Starting with the first column of S, we select any nonzero element, say s_{i1}, (selecting the element with the largest absolute value will be advantageous computationally), for use as a pivot in the sense of the following calculation

$$\bar{s}_1 = s_1 - e_i \tag{4}$$

$$S^{(1)} = S - (s_{i1})^{-1} \, \bar{s}_1 (e_i^T \, S) \tag{5}$$

where e_k represents the k^{th} unit column vector of appropriate dimension, and the superscript T represents transposition.

In words, the calculation may be stated as follows. Locate any nonzero element s_{i1} in the first column, s_1, of S. Then $\bar{s}_1$ is obtained by subtracting one from the i^{th} element, s_{i1}, of s_1. Take the outer product of $\bar{s}_1$, scaled by $(s_{i1})^{-1}$, and the i^{th} row of S. Subtracting this outer product from S gives $S^{(1)}$.

It is easily verified that $S^{(1)}$ is a nonsingular transformation of S and that the first column of $S^{(1)}$ is e_i. This process is now repeated for columns 2, 3, etc., to get a different unit vector in each case, and skipping a column if it is a linear combination of the unit vectors to its left. Since the rank of S is n, after a maximum of n steps, $S^{(n)}$ will contain n different unit column vectors, in addition to other columns, which are linear combinations of the unit vectors to their left. Permutation of the rows, so that the unit vectors appear in their natural order, gives the Hermite normal form of the matrix

$$H = [e_1 \; e_2 \; \ldots \; e_{n_1} \; \alpha_1 \; * \; * \; \ldots \; * \; \ldots \; e_n \; \alpha_p \; * \; * \; \ldots] \tag{6}$$

where the asterisks represent columns which do not matter for our purpose.

We may now identify the following canonical form matrices

$$\bar{A} = [e_2\ e_3 \ldots e_{n_1}\ \alpha_1 \ldots\ldots e_n\ \alpha_p] \tag{7}$$

$$\bar{B} = [e_1\ e_{n_1+1}\ e_{n_1+n_2+1} \ldots e_{n-n_p+1}] \tag{8}$$

$$\bar{C} = C\ P = C[b_1\ Ab_1 \ldots A^{n_1-1}\ b_1 \ldots b_p\ Ab_p \ldots A^{n_p-1}\ b_p] \tag{9}$$

It may be noted that the columns of the transformation matrix P are obtained from the columns of S, and no matrix inversion has been necessary. Since the non-unit column vector in each block is a linear combination of the unit vectors preceding it (including those in the blocks to the left), the matrix $\bar{A}$ will be of the block upper triangular form. It may be pointed out that sometimes the canonical matrix $\bar{A}$ will have fewer than p blocks, say r. This simply implies that the system is controllable from only the first r inputs. The resulting B will, therefore, have unit vectors in its first r columns only; the remaining p-r columns will be of no special form.

3. TRANSFORMATION TO THE ROW-COMPANION FORM

Since the row-companion form is the dual of the column-companion form, the procedure for transformation is similar to that described in the previous section. The essential difference is that this time one proceeds from the rows of the observability matrix, and the transformation to the Hermite normal form is performed on a row basis.

Define

$$T = \begin{bmatrix} c^1 \\ c^1 A \\ \vdots \\ c^1 A^n \\ c^2 \\ \vdots \\ c^2 A^{n-1} \\ \vdots \\ c^q A^{n-q+1} \end{bmatrix} = \begin{bmatrix} t^1 \\ t^2 \\ \vdots \\ t^{n+1} \\ t^{n+2} \\ \vdots \\ \vdots \\ t_{nq+n-\frac{1}{2}q^2+\frac{1}{2}q} \end{bmatrix} \tag{10}$$

where c^i is the i^{th} row of C.

Starting with the first row of T, we select a nonzero pivot, say t_i, and then proceed as below

$$T^{(1)} = T - (t_{1i})^{-1} (Te_i) (t^1 - e^i) \tag{11}$$

where e^i is the i^{th} unit row vector of appropriate dimension.

This process is now repeated for rows 2, 3, etc., to get a different unit row vector in each case, skipping a row if it is a linear combination of the unit row vectors above it. After a maximum of n steps, $T^{(n)}$ will contain n different unit row vectors in addition to other rows which are linear combinations of the unit row vectors above them. Permutation of the columns so that the unit row vectors appear in there natural order gives the Hermite normal form of the matrix, for which the canonical form matrices are obtained in a manner similar to that in the previous section.

4. TRANSFORMATION TO THE INPUT-IDENTIFIABLE FORM

The transformation to the input-identifiable form is obtained in a manner similar to the procedure described in section 2 for obtaining the column-companion form. The main difference is that now one starts with the matrix

$$\tilde{S} = [B \quad AB \quad A^2B \ \dots \ A^{n-p+1} \quad B] \tag{12}$$

Transforming it to the Hermite normal form on the column basis, and interchanging the rows to get the unit column vectors in the natural order, one has

$$\tilde{H} = [\tilde{b}_1 \ \tilde{b}_2 \ \dots \ \tilde{b}_p \quad \tilde{A}\tilde{b}_1 \quad \tilde{A}\tilde{b}_2 \ \dots \ \tilde{A}^{n-p+1} \ \tilde{b}_p] \tag{13}$$

In the generic case, when the inputs are linearly independent, the vector $\tilde{b}_i$ will be the unit vector e_i, $i=1,2,\dots,p$. The input identifiable form is then obtained as

$$\tilde{B} = [\tilde{b}_1 \quad \tilde{b}_2 \ \dots \ \tilde{b}_p] = [e_1 \quad e_2 \ \dots \ e_p] \tag{14}$$

$$\tilde{A} = [\text{The next n columns of H}] \tag{15}$$

and

$$\tilde{C} = C \ \tilde{P} \tag{16}$$

where the transformation matrix $\tilde{P}$ is obtained from the n columns of $\tilde{S}$ corresponding to the unit vecotrs of $\tilde{H}$, and in the same order.

5. TRANSFORMATION TO THE OUTPUT-IDENTIFIABLE FORM

In this case, the procedure is similar to that outlined in section 3, with the difference that now one starts with the matrix

$$\tilde{T} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-q+1} \end{bmatrix} \tag{17}$$

It is then transformed to the Hermite normal form so as to obtain n unit row vectors occuring in their natural order, with permutation of columns if necessary. The canonical form matrices are then obtained following a procedure analogous to that in the previous section.

6. TRANSFORMATION TO THE CONTROLLABLE CANONICAL FORM

This transformation requires two steps. The first step is to obtain the input-identifiable form for the system, following the procedure described in section 4. From the n unit vectors in $\tilde{H}$, one then selects the p unit column vectors

$$q_i = \tilde{A}^{n_i - 1} \tilde{b}_i, \ i = 1, 2, \ldots, p \tag{18}$$

where n_i is the maximum possible integer for which the right-hand side is a unit vector and thus, independent of the vectors to its left. The integers n_i, i=1,2,...,p are then the controllability subindices of the system, with

$$\sum_{i=1}^{p} n_i = n \tag{19}$$

Defining the transpose of q_i as the unit row vector q^i, one then generates the augmented transformation matrix Q given by

$$Q = \begin{bmatrix} q^1 \\ q^1\tilde{A} \\ \vdots \\ q^1\tilde{A}^{n_1-1} \\ \hline q^2 \\ q^2\tilde{A} \\ \vdots \\ q^p\tilde{A}^{n_p-1} \\ \hline q^1\tilde{A}^{n_1} \\ q^2\tilde{A}^{n_2} \\ \vdots \\ q^p\tilde{A}^{n_p} \\ \hline \tilde{C} \end{bmatrix} \tag{20}$$

Transforming Q to the Hermite normal form on a row basis, following the procedure described in section 3, and rearranging the columns to obtain the unit row vectors in their natural order, one obtains

$$\bar{H} = \left[\begin{array}{c} I_n \\ \hdashline \alpha^1 \\ \alpha^2 \\ \vdots \\ \alpha^p \\ \hdashline \bar{C} \end{array}\right] \tag{21}$$

where I_n is an nxn identity matrix.

The matrices for the controllable canonical form are then obtained as

$$\bar{A} = \left[\begin{array}{c} e^2 \\ e^3 \\ \vdots \\ e^{n_1} \\ \alpha^1 \\ \hdashline e^{n_1+2} \\ \vdots \\ \alpha^2 \\ \hdashline \vdots \\ e^n \\ \alpha^p \end{array}\right] \tag{22}$$

$$\bar{B} = Q_{np}\text{; the first n rows and p columns of Q} \tag{23}$$

and $$\bar{C} = \text{the last q rows of H} \tag{24}$$

It may be noted that a nonsingular transformation of the input may be required to make each nonzero row of B a unit vector [1].

7. TRANSFORMATION TO THE OBSERVABLE CANONICAL FORM

The procedure in this case is the dual of that described in the previous section. The state equations of the system are first transformed to the output identifiable form as described in section 5. From $\tilde{H}'$, $\tilde{A}$ and $\tilde{C}$, one then selects the q unit row vectors

$$r^i = \tilde{C}^i \tilde{A}^{n_i-1}, \quad i = 1, 2, \ldots, q \tag{25}$$

where n_i is the largest possible integer for which the right-hand side is a unit vector. Defining the transpose of r^i as the unit column vector r_i, one then generates the matrix

$$R = [r_1 \; Ar_1 \; \ldots \; A^{n_1-1} r_1 \; r_2 \; \ldots \; A^{n_q-1} r_q \quad A^{n_1} r_1 \; \ldots \; A^{n_q} r_q \quad B] \tag{26}$$

Transformation of R to the Hermite normal form on the column basis gives

$$\bar{H}' = [I_n \quad \beta_1 \; \beta_2 \; \ldots \; \beta_q \quad \bar{B}] \tag{27}$$

Hence, the observable canonical form is obtained as

$$\bar{A} = [e_2 \; e_3 \; \ldots \; e_{n_1} \quad \beta_1 \; e_{n_1+2} \; \ldots \; e_{n_1+n_2} \quad \beta_2 \; \ldots \; e_n \quad \beta_q] \tag{28}$$

and $\bar{C} = R_{qn}$; the first q rows and n columns of R (29)

8. EXAMPLE

Consider the transformation of the following system to the controllable canonical form

$$A = \begin{bmatrix} 28 & 0 & -7 & 13 & 128 \\ -4 & 0 & 0 & -2 & -20 \\ 7 & -1 & -5 & 3 & 27 \\ 10 & 11 & -6 & 2 & 34 \\ -7 & -1 & 2 & 3 & -31 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & -6 \\ 0 & 1 \\ 0 & -2 \\ -2 & 2 \\ 0 & 1 \end{bmatrix}$$

and

$$C = \begin{bmatrix} 0 & 6 & 4 & 0 & 3 \\ 3 & 5 & -1 & 1 & 11 \end{bmatrix}$$

First it is transformed to the input-identifiable form using the matrix

$$\tilde{S} = [B \quad AB \quad A^2B \quad A^3B \quad A^4B]$$

$$= \begin{bmatrix} 1 & -6 & 2 & 0 & -1 & 13 & -4 & -15 & 20 & -33 \\ 0 & 1 & 0 & 0 & 0 & -2 & 0 & 4 & 0 & -4 \\ 0 & -2 & 1 & 0 & 0 & 3 & -4 & -1 & 16 & -40 \\ -2 & 2 & 6 & 1 & -8 & 2 & 8 & -8 & 0 & -4 \\ 0 & 1 & -1 & 0 & 1 & 3 & 0 & 4 & -4 & 7 \end{bmatrix}$$

Selecting the first element in the first column as the pivot, we have

$$\tilde{S}^{(1)} = \tilde{S} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ -2 \\ 0 \end{bmatrix} [1 \quad -6 \quad 2 \quad 0 \quad -1 \quad 13 \quad -4 \quad -15 \quad 20 \quad -33]$$

$$= \begin{bmatrix} 1 & -6 & 2 & 0 & -1 & 13 & -4 & -15 & 20 & -33 \\ 0 & 1 & 0 & 0 & 0 & -2 & 0 & 4 & 0 & -4 \\ 0 & -2 & 1 & 0 & 0 & 3 & -4 & -1 & 16 & -40 \\ 0 & -10 & 10 & 1 & -10 & -24 & 0 & -38 & 40 & -70 \\ 0 & 1 & -1 & 0 & 1 & 3 & 0 & 4 & -4 & 7 \end{bmatrix}$$

Proceeding in this manner, the Hermite normal form, after permutation of rows, is obtained as

$$\tilde{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & -2 & 0 & 2 \\ 0 & 1 & 0 & 0 & 0 & -2 & 0 & 4 & 0 & -4 \\ 0 & 0 & 1 & 0 & 0 & -1 & -4 & 11 & 16 & -48 \\ 0 & 0 & 0 & 1 & 0 & -2 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -2 & -4 & 11 & 12 & -37 \end{bmatrix}$$

The input-identifiable form is then given by

$$\tilde{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -2 & 0 \\ 1 & 0 & 0 & -1 & -4 \\ 0 & 1 & 0 & -2 & 0 \\ 0 & 0 & 1 & -2 & -4 \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\tilde{C} = \begin{bmatrix} 0 & 6 & 4 & 0 & 3 \\ 3 & 5 & -1 & 1 & 11 \end{bmatrix} \begin{bmatrix} 1 & -6 & 2 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & -2 & 1 & 0 & 0 \\ -2 & 2 & 6 & 1 & -8 \\ 6 & 1 & -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 0 & 3 \\ 1 & 2 & 0 & 1 & 0 \end{bmatrix}$$

From $\tilde{H}$, $\tilde{A}$ and $\tilde{B}$, it is noted that $q_1 = e_5$, $q_2 = e_4$, $n_1 = 3$ and $n_2 = 2$. Hence,

$$Q = \left[\begin{array}{c} q^1 \\ q^1\tilde{A} \\ q^1\tilde{A}^2 \\ \hline q^2 \\ q^2\tilde{A} \\ \hline q^1\tilde{A}^3 \\ q^2\tilde{A}^2 \\ \hline \tilde{C} \end{array}\right] = \left[\begin{array}{ccccc} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & -2 & -4 \\ 1 & -2 & -4 & 11 & 12 \\ \hline 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & -2 & 0 \\ \hline -4 & 11 & 12 & -37 & -32 \\ 0 & -2 & 0 & 2 & 0 \\ \hline 0 & 1 & 1 & 0 & 3 \\ 1 & 2 & 0 & 1 & 0 \end{array}\right]$$

While transforming Q to the Hermite normal form on the row basis, advantage may be taken of the fact that p rows of Q are already unit vectors. In this example, $q^1 = e^5$ and $q^2 = e^4$. Hence, we need only transform the second, third and fifth rows of Q to unit vectors. The Hermite normal form, after permutations of columns, is obtain as

$$\overline{H} = \left[\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ \hline 0 & -4 & -4 & 5 & 3 \\ 0 & 0 & 0 & -2 & -2 \\ \hline 7 & 1 & 0 & 4 & 1 \\ 4 & 4 & 1 & 6 & 4 \end{array}\right] = \left[\begin{array}{c} e^1 \\ e^2 \\ e^3 \\ \hline e^4 \\ e^5 \\ \hline \alpha^1 \\ \alpha^2 \\ \hline c^1 \\ c^2 \end{array}\right]$$

The controllable canonical form matrices are then obtained as

$$\overline{A} = \left[\begin{array}{ccccc} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -4 & -4 & 5 & 3 \\ \hline 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -2 & -2 \end{array}\right], \quad \overline{B} = \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 1 & -2 \\ \hline 0 & 0 \\ 0 & 1 \end{array}\right]$$

$$\overline{C} = \begin{bmatrix} 7 & 1 & 0 & 4 & 1 \\ 4 & 4 & 1 & 6 & 4 \end{bmatrix}$$

It may be noted that $\overline{B}$ is still not in the desired form, and a nonsingular transformation of the input is necessary to give

$$\overline{B}' = \overline{B}\, M = \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ \hline 0 & 0 \\ 0 & 1 \end{array}\right]$$

where

$$M = \begin{bmatrix} 1 & -2 \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}$$

9. CONCLUSIONS

An efficient algorithm has been presented for the transformation of the state equations of a linear multivariable system to any of six canonical forms. The transformations do not require separate determination of a set of linearly independent vectors from a rectangular matrix, nor do they require matrix inversion. It may be added that the authors have a working computer program, based on the methods described in this paper, which can be utilized for the transformations.

10. REFERENCES

1. N.K. Sinha and P. Rózsa, "Some canonical forms for linear multivariable systems", International Journal of Control, vol. 23, 1976, pp. 865-883.

2. D.G. Luenberger, "Canonical forms for linear multivariable systems", IEEE Trans. on Automatic Control, vol. AC-12, 1967, pp. 290-293.

3. P. Rózsa and N.K. Sinha, "Efficient algorithm for irreducible realization of a rational matrix", International Journal of Control, vol. 20, 1974, pp. 739-751.

4. P. Rózsa and N.K. Sinha, "Minimal realization of a transfer function matrix in canonical forms", ibid., vol. 21, 1975, pp. 273-284.

5. D. Jordan and B. Sridhar, "An efficient algrotihm for calculation of the Luenberger canonical form", IEEE Trans. on Automatic Control, vol. AC-18, 1973, pp. 292-295.

6. J.D. Aplevich, "Direct computation of canonical forms for linear systems by elementary matrix operations", ibid., vol. AC-19, 1974, pp. 124-125.

THEORY OF CONE METHODS FOR NULL-CONTROLLABILITY OF NONLINEAR CONTROL SYSTEMS

A. S. C. SINHA
School of Engineering and Technology
IU-PU, Indianapolis, IN 46202

Abstract. Cone-valued Lyapunov functions have been used by several authors to examine stability of nonlinear systems. However, the general problem of constructing suitable cone given the nature of nonlinearity seems to be as difficult as that of constructing a suitable Lyapunov function. The construction of suitable cone at least in special cases deserves further investigation. Lakshimikanthan, V. [1] has used theory of cone for stability of nonlinear systems. A more unified theory of cones can be found in the book by Krasnosel'skii, M. A. [2]. The objective of this paper is to use methods of theory of cones to establish a sufficient condition of null controllability of $\dot{x}(t) = A(t)\ x(t) + g(t,x,u)$.

INTRODUCTION

Consider the nonlinear control system

$$\dot{x}(t) = A(t)x + g(t,x,u) \tag{1}$$

where $g(t,x,u) \in C'(R^{n+m})$ for each $t \in I_0$. It is assumed throughout the sequel u belongs to the set of all bounded measurable control functions with values $u(t) \in \Omega \subset R^m$. Such controls will be called admissible. Next we give several definitions.

Definition 1 (Cone): A proper subset K of R^n is called cone if (i) $\lambda K \subset K$, $\lambda \geq 0$, (ii) $K + K \subset K$, (iii) $K = \overline{K}$, (iv) $K \cap (-K) = \{0\}$ and (iv) K^0 is nonempty. Here $\overline{K}$ denotes the closure of K, and K^0 the interior of K.

Definition 2 (Order Relations): The cone K induces the order relations on R^n defined by $x \underset{K}{\leq} y$ iff $y - x \in K$ and $x \underset{K^0}{<} y$ iff $y - x \in K^0$.

For definitions 1 and 2, see Krasnosel'skii [8] and Lakshmikantham [3]. We will now give some definitions on controllability which have appeared in several forms. Our definitions are similar to Chukwu [4].

Definition 3 (Locally Controllable to the Origin): Define the domain ζ of null controllability of (1) to be the set of all initial points x_0 in R^n such that u(t) is an admissible control and x(t) a solution of (1) with $x(t_1) = 0$ for some finite time $t_1 \geq t_0$. If ζ contains an open neighborhood of zero then (1) is said to be locally controllable to the origin.

Definition 4 (Reachable Set): The reachable set of (1) at time t is the set

$$IR(t,t_0) = \int_{t_0}^{t} X^{-1}(s,t_0)\ Q(s)ds;\ Q(t) = B(t)\ u(t),\ u \in \Omega$$

where $X(t,t_0)$ is the transition matrix of $\dot{x}(t) = A(t)x(t)$.

Definition 5 (Proper): The system (1) is proper on $[t_0,t]$, $t > t_0$ if and only if the origin is an interior point of $IR(t,t_0)$.

Definition 6 (Locally Controllable to the Origin): If $0 \in \Omega$, and the related linear system is proper, then (1) is locally controllable to the origin.

The objective of this paper is to use cone methods to analyze the null controllability of a nonlinear perturbed system. Sufficient conditions on g are placed which guarantee that the domain of null controllability of (1) is the whole Euclidean space R^n.

Consider the nonlinear control system

$$\dot{x}(t) = A(t)x + g(t,x,u)$$

where we assume that A(t) is of the form

$$A(t) = \begin{bmatrix} 0 & 1 & 0, & \dots, & 0 \\ 0 & 0 & 1, & \dots, & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ -A_n(t), & -A_{n-1}(t), & & \dots, & -A_1(t) \end{bmatrix} \tag{2}$$

where $A_n(t), \dots, A_1(t)$ are continuous functions on $[t_0\infty)$. The characteristic polynomial of (2) is given by

$$\Delta(t,\lambda) = \lambda^n + A_1(t)\lambda^{n-1} + \cdots + A_n(t). \tag{3}$$

Let $\lambda_1, \lambda_2, \dots, \lambda_n$ be certain constants. In order to construct a cone K_0, we define polynomials $T_n(t,\lambda_1, \dots, \lambda_{n-1})$ as follows:

$$T_n(t,\lambda_1, \dots, \lambda_{n-1}) = \prod_{i=1}^{n-1} (\lambda - \lambda_i)$$

$$= \lambda^{n-1} + t_{n,n-1}\lambda^{n-1} + \cdots + t_{n1} \tag{4}$$

Now construct an operator T as

$$T = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ t_{21} & 1 & 0 & & \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ t_{n1} & t_{n2} & t_{n3} & \dots & 1 \end{bmatrix} \tag{5}$$

Definition 7: The cone K^+ is defined by

$$K^+ = \{u: \ Tu^+ \geq 0\}$$

We will now state some lemmas.

Lemma 1: Let $\lambda_1, \dots, \lambda_{n-1}$ exist such that $\Gamma^K(t,\lambda_1,\dots, \lambda_{n-1}) \leq 0$, $K = 0,1,\dots,n-2$, and then for some $\lambda_0 > \max \lambda_i$ $(1 \leq i \leq n-1)$ and $\Delta(t,\lambda_0) \geq 0$, the following estimate for the system $\dot{x}(t) = A(t)\, x(t)$ holds

$$||x(t,\tau)|| \leq C\, e^{\lambda_0(t-\tau)}; \quad C > 0, \quad t \geq 0$$

where $x(t,\tau)$ is the fundamental matrix such that $x(t,t) = I$, and the polynomial $\Gamma^K(t,\lambda_1,\dots,\lambda_{n-1})$ is defined subsequently.

The proof is based on theory of cones methods.

For the autonomous system in R^n of type

$$\dot{x} = Ax + Bu + g(x,u) \tag{6}$$

where g together with its partial derivatives are continuous in R^{n+m} and zero at the origin in R^{n+m}, and where the admissible controls are assumed continuous, Lukes [5] proved the following result:

Lemma 2. If for the control system (6) in R^n

(i) rank $[B, AB, \ldots, A^{n-1}B] = n$

(ii) g together with its partial derivatives are continuous in R^{n+m}, and zero at the origin in R^{n+m}

(iii) $||g(x,u)||$ is bounded in R^{n+m}

then the domain of null controllability of (6) is R^n.

We now state our main results.

Theorem 1. Let conditions of Lemma 1 hold and if for some admissible control u^*

(i) $\lambda_0 \leq \delta < 0$ for all $(x,t) \in R^n \times I_0$

(ii) $||g(t,x,u^*)|| \leq \beta_1(t)\,||x|| + \beta_2(t)\,||x||^{\beta}$

for all $(x,t) \in R^n \times I_0$, and $\beta_1(t) \geq 0$, $\beta_2(t) \geq 0$ are continuous functions of t and β is in the range $0 \leq \beta \leq 1$.

(iii) $\int_{\tau}^{t} \beta_1(s)ds \leq \mu_1(t - \tau) + \mu_2;$

$\mu_1 \geq 0, \quad \mu_2 \geq 0, \quad t \geq \tau$

(iv) $\int_{t}^{t+r} ||\beta_2(s)||^p \, ds \to 0 \quad$ as $\quad t \to \infty$

for some $r \geq 0$, and $1 \leq p \leq 2$.

(v) $0 \in \Omega$, and $g(0,0,t) = 0$, and the linear system is proper.

Then the domain ζ of null controllability of (1) is R^n.

Remark: Chukwu [4] and Sinha [6] have used Lyapunov methods to examine the null controllability of control systems. In this paper we have constructed a cone K^+ and used theory of cone methods to analyze the problem. Although constructing cone K^+ is quite difficult, it appears that for a certain class of systems, theory of cone methods can generalize the existing results.

Proof of Lemma 1. Let $\lambda_1, \lambda_2, \ldots, \lambda_{n-1}$ are constants satisfying eqn. 3, and let λ_0 be any constant such that $\lambda_0 > \max \lambda_i$ $(1 \leq i \leq n-1)$. We introduce the following notation

$$\Gamma^0(t,\lambda_1) = \Delta(t,\lambda_1), \quad \Gamma^1(t,\lambda_1,\lambda_2) = \frac{\Delta(t,\lambda_1) - \Delta(t,\lambda_2)}{\lambda_1 - \lambda_2}$$

$$\Gamma^2(t,\lambda_1,\lambda_2,\lambda_3) = \frac{\Gamma^1(t,\lambda_1,\lambda_2) - \Gamma^1(t,\lambda_1,\lambda_3)}{\lambda_2 - \lambda_3}$$

$$\Gamma^3(t,\ \lambda_1,\ \ldots,\ \lambda_4) = \frac{\Gamma^2(t,\lambda_1,\lambda_2,\lambda_3) - \Gamma^2(t,\lambda_1,\lambda_2,\lambda_4)}{\lambda_3 - \lambda_4}$$

. . . .

$$\Gamma^{n-2}(t,\ \lambda_1,\ \ldots,\ \lambda_{n-1}) = \frac{\Gamma^{n-3}(t,\lambda_1,\ldots,\lambda_{n-2}) - \Gamma^{n-3}(t,\lambda_1,\ldots,\lambda_{n-3},\lambda_{n-1})}{\lambda_{n-2} - \lambda_{n-1}}$$

If some λ_i are equal, the expression given above should be considered in the limiting form. We transform the linear system $\dot{x} = A(t)x$ by a substitution $y = T\ x$ (T is defined in eqn. 5) to

$$\dot{y} = T\ A\ T^{-1}\ y$$

A simple computation shows that

$$TAT^{-1} = \left[\begin{array}{ccccc} \lambda_1 & 1 & 0 & \cdots & 1 \\ 0 & \lambda_2 & 1 & & \\ \cdot\ \cdot\ \cdot & \cdot\ \cdot\ \cdot & \cdot\ \cdot\ \cdot & \cdot\ \cdot\ \cdot & \cdot\ \cdot\ \cdot \\ 0 & 0 & 0 & & 1 \\ -\Gamma^0(t,\lambda_1) & -\Gamma^1(t,\lambda_1,\lambda_2) & -\Gamma^2(t,\lambda_1,\lambda_2,\lambda_3), & \ldots, & -\Gamma^{n-1}(t,\lambda_1,\ldots,\lambda_{n-1} \end{array}\right]$$

where we let $\Gamma^{n-1} = -A(t) - \lambda_i -,\ \ldots,\ -\lambda_{n-1}$. Under the assumption that $\Gamma^K(t,\lambda_1,\ldots,\lambda_{n-1}) \leq 0$, $(K = 0,1,\ldots\ \ )$ the cone K^+ remains invariant under action of $x(t',\tau)$. $x(t,\tau)$ is fundamental matrix of the system $\dot{x} = A(t)\mathbf{x}$ such that $x(t,t) = I$. For any set $\mathbf{x}_0 = (1,\lambda_0,\ldots,\lambda_0^{n-1})$, it follows that $T\mathbf{x}_0 = (1,\Delta_2(\lambda_0),\ \ldots,\ \Delta_n(\lambda_0)]$ where Δ_k's are defined in eqn. 3, then we can introduce the u_0-norm in space R^n in the sense of ref. 2. Let $||u||_{u_0} = \min u$.

Now consider the function $x_0(t,\tau) = e^{\lambda_0(t-\tau)}$ so that $\frac{dx_0(t,\tau)}{dt} \geq A(t)x_0(t,\tau)$; $(t \geq \tau \geq t_0)$, since $\lambda_0 > \max \lambda_i$ and $\Delta(t,\lambda_0) \geq 0$. Following results of ref. 2

$$-e^{\lambda_0(t-\tau)}x_0 \overset{0}{\leq} \phi(t,\tau)x \overset{0}{\leq} e^{\lambda_0(t-\tau)}x_0$$

whenever order relation $-x_0 \overset{0}{\leq} x \overset{0}{\leq} x_0$. Therefore,

$$||x(t,\tau)||_{u_0} \leq e^{\lambda_0(t-\tau)}; \quad (t \geq \tau \geq t_0)$$

which concludes the proof of lemma 1.

Proof of Theorem 1. It is well known that if $||y(t,\tau)|| \leq C\ e^{\lambda_0(t-\tau)}$, $C > 0$, $\lambda_0 < 0$, then there exists a Lyapunov function $v(y,t) = (Py(t),\ y(t))$ such that $C||y(t)||^2 \geq v(t) \geq b||y(t)||^2$, and whose derivative along the trajectories of $\dot{y}(t) = T\,A\,T^{-1}y$ satisfy the inequality $\dot{v}(t) \leq -2\delta_0 v$. Now consider the transformed eqn. 1

$$\dot{y} = TAT^{-1}y + T\,g\ (t,\ T^{-1}y,\ u^*) \tag{7}$$

Therefore $\dot{v}(t)$ along the trajectories of (7) gives

$$\dot{v}(t) \leq -2\delta_0 v + D_1\left(\beta_1(t)\ ||y||^2 + \beta_2(t)\ ||y||^{1+\beta}\right) \equiv -2\delta_0 v + w(t)$$

because $||T\,g\ (t,\ T^{-1}y,\ u^*)|| \leq D_0\left(\beta_1(t)\ ||y|| + \beta_2(t)\ ||y||^{\beta}\right)$ for some $D_0 > 0$, since assumption (ii) holds. Using bounds on $v(t)$, we have

$$w(t) \leq D_2\left[\beta_1(t)v + \beta_2(t)v^{\alpha_1}\right]; \quad \alpha_1 = \tfrac{1}{2}(\beta + 1)$$

$$D_2 = D_1 \max(C^{-1}, C^{-\alpha_1}).$$

It follows that

$$\begin{aligned}\dot{v}(t) &\leq -(\delta_0 - D_2\beta_1(t))v + (-\delta_0 v + D_2\beta_2(t)v^{\alpha_1}) \\ &\leq -(\delta_0 - D_2\beta_1(t))v + 2Dv^{\mu}[\beta_2(t)]^{p/1-\beta} \qquad (8)\end{aligned}$$

Since the inequality $-\delta_0 v + D_2\beta_2(t)v^{\alpha_1} \leq 2Dv^{\mu}[\beta_2(t)]^{p/(1-\beta)}$ can be seen to hold for $1 \leq p \leq 2$ and $\mu = 1 - \frac{1}{2}p$, for some $D > 0$. The arguments in proving the inequality follows similar to Chukwu [4]. On substituting $v = w^{2/p}$ in eqn. 8, and on integrating the resulting differential inequality we obtain for $t \geq t_0$

$$v^{\frac{1}{2}p} = v^{\frac{1}{2}p}(t_0)\exp\left\{-\tfrac{1}{2}p\ \phi(t,t_0)\right\} + pD\int_{t_0}^{t}[\beta_2(\tau)]^{p/1-\beta}\exp\left\{-\tfrac{1}{2}p\ \phi(t,\tau)\right\}d\tau \qquad (9)$$

where $$\begin{aligned}\phi(t,\tau) &= \int_{\tau}^{t}\left\{\delta_0 - D_2\beta_1(t)\right\}dt \\ &\geq (\delta_0 - D_2\mu_1)(t - \tau) - D_2\mu_2,\end{aligned}$$

so that if we further assume that $\mu_1 \leq \frac{1}{2}\ \delta_0\ D_2^{-1}$ then $\phi(t,\tau) \geq \frac{1}{2}\delta_0(t-\tau) - D_2\mu_2$. Then eqn. 9 is rewritten as

$$\begin{aligned}v^{\frac{1}{2}p} &\leq v^{\frac{1}{2}p}(t_0)\exp\left\{-\tfrac{1}{2}p\delta_0(t-\tau)\right\} \\ &\quad + pC\int_{t_0}^{t}[\beta_2(\tau)]^{p/(1-\beta)}\exp\left\{-\tfrac{1}{2}p\ \delta_0\ (t-\tau)\right\}d\tau \qquad (10)\end{aligned}$$

for all $t \geq t_0$. Utilizing bounds on $v(t)$ once again, we see that the above inequality implies that

$$\begin{aligned}\|x(t)\|^{p} &\leq C_1\|x(t_0)\|^{p}\exp\left\{-\tfrac{1}{2}p\ \delta_0\ (t - t_0)\right\} \\ &\quad + C_2\int_{t_0}^{t}[\beta_2(\tau)]^{p/(1-\beta)}\exp\left\{-\tfrac{1}{2}p\ \delta_0\ (t - \tau)\right\}d\tau \qquad (11)\end{aligned}$$

Since the above inequality is similar to Chukwu, we omit our arguments which follows exactly as §3 of ref. [4]. Therefore, there is a finite time T such that every solution $x(t)$ with $x(t_0) = x_0$ has $x(T)$ in the neighborhood of the origin. By our assumption, the linear system is proper, and hence $x(T)$ is indeed steered to the origin in finite time by some admissible control.

REFERENCES

1. Lakshmikantham, V., Vector Lyapunov functions, Proceedings of Twelfth Annual Allerton Conference on Circuit and System Theory, pp. 71-77, (1974).

2. Krasnosel'skii, M. A., Positive Solutions of Operator Equations (Translated from the Russian); P. Noordhoff, Ltd., Groningen, The Netherlands, (1964).

3. Lakshmikantham, V., Cone-Valued Lyapunov Functions, Nonlinear Analysis, Theory, Methods and Applications, vol. 1, No. 3, pp. 215-222, (1977).

4. Chukwu, E. N., Finite Time Controllability of Nonlinear Control Processes, SIAM J. Control, Vol. 13, (1975).

5. Lukes, D. L., Global Controllability of Nonlinear Systems, SIAM J. Control, 10, pp. 112-126, (1972).

6. Sinha, A. S. C., Null Controllability of AMCD-Spacecraft Attitude Control System, IEEE - Southeastcon 77, Williamsburg, pp. 602-608.

7. Krasnosel'skii, M. A., The Operator of Translation along the Trajectories of Differential Equations, Translation of Mathematical Monographs, Vol. 19, AMS, Rhode Island, 1968.

THE GENERATION OF PROGRAMMED LOGIC ARRAY (PLA) IMPLEMENTATIONS FROM HIGHER LEVEL LANGUAGE DESCRIPTIONS

J. W. GAULT
Electrical Engineering Department
North Carolina State University
Raleigh, NC 27607

Abstract

The concept of programmed logic design is most often associated with microcomputer-based solutions and the power of programmed logic arrays is frequently overlooked. This paper describes the basic concepts of a compiler which has been written to accept a higher level language (PL/M[1]) programmed logic problem description and for certain restricted classes of problems, produce the code required for a PLA implementation. The speed characteristics of the PLA hardware solution are significantly faster than the corresponding microcomputer solution and the desirable features of language-based design are retained.

INTRODUCTION

A *programmed logic array* (PLA) is a large scale integrated package which contains an array of logic gates capable of realizing a large number of logic equations in two-level sum-of-products form [1,2,5]. The specific behavior of a PLA is established by defining the connections in the AND and OR arrays according to the literals which appear in the desired equation set. The objective of this paper is to show how the connections which are required may be derived by analyzing a high level language description of the problem solution and thus avoid the generation of detailed equations. The sections that follow describe the nature of the problem description limitations and the form of the compiled results. An example is used to demonstrate the utility of the procedure.

DESIGN LANGUAGE

The PL/M language was selected since it has considerable acceptance as a higher level language in microcomputer-based systems. The class of problems which are suitable for the compiler described here are control problems with the following characteristics:

---

[1] Intel Corporation

- decision elements restricted to single product terms or the OR of single literals,
- structured flow control utilizing a block structure with DOWHILE and IF THEN ELSE statements and
- terminal tasks which reduce to the activation of pulse or level signals. Multiple output tasks may be defined concurrently with the notation ACTION1*ACTION2.

The most noteworthy characteristics which are typically found in microcomputer-based systems and not allowed here are arithmetic computations and subroutine linkage. The present structure does allow the use of MACRO functions which are restricted in all of the ways listed above. Many classic digital operations (SHIFT, COUNT, DELAY) have well defined PLA solution configurations which can be readily cataloged and utilized by the compiler [4].

Example 1 shows a language description for a simple traffic light controller with a turn light which is activated on demand. Each green light is left on for a different amount of time (unbound here), while the amber and red are all equal fixed times.

Example 1

```
/*LIGHT SEQUENCE CONTROL*/
/*NS IS NORTH/SOUTH LANE, EW IS EAST/WEST, AND T IS TURN LANE*/
   SET(NS RED)*SET(EW RED)*SET(T RED);
DO FOREVER;
   SET(NS GREEN)*CLEAR(NS RED);
   DELAY(NS G);
   SET(NS AMBER)*CLEAR(NS GREEN);
   DELAY(4);
   SET(NS RED)*CLEAR(NS AMBER);
   DELAY(1);
   SET(EW GREEN)*CLEAR(EW RED);
   DELAY(EW G);
   SET(EW AMBER)*CLEAR(EW GREEN);
   DELAY(4);
   SET(EW RED)*CLEAR(EW AMBER);
   DELAY(1);
   IF (SENSOR1+SENSOR2)
      THEN DO SET(T GREEN)*CLEAR(T RED);
              DELAY(T G);
              SET(T AMBER)*CLEAR(T GREEN);
              DELAY(4);
              SET(T RED)*CLEAR(T AMBER);
              DELAY(1);
           END
      ELSE;
END;
```

The SET and CLEAR are activation signals which control FFs associated with each light. The DELAY(x) is a macro used to provide a delay between light changes.

COMPILED DESIGN

The compiler accepts a language description like that shown in example 1, checks it for proper syntax, assigns variable names to PLA inputs and outputs, and produces a string which can be reduced to PLA code [7]. The post processor which analyzes this string must assign states and output values in terms of PLA array connections. The resulting hardware (Fig. 1) looks like a conventional sequential machine with FF state variables and feedback. In Fig. 1 a total of five state variables are required with an additional six FFs provided for the DELAY MACRO. Since all of the system outputs are levels rather than pulses they are latched into an additional nine FFs. The ACK feedback is a synchronization signal from the MACRO function.

The mapping rule, which is applied by the post processor to assign states, will generate a new state for every

- DO-WHILE entry point,
- IF-THEN-ELSE entry point which has an ACTION as any immediate predecessor, and
- ACTION entry point which begins a procedure or which has another ACTION as an immediate predecessor.

This assignment procedure is straightforward to implement and it produces reasonable results, not necessarily the minimal number of states, however.

In the PLA solution environment the number of states is important in that the number of feedback variables, and hence PLA inputs,and outputs, are impacted. The number of product terms required by a solution is often more binding than the I/O pin limitations. Each state will generate a product term (one PLA entry) for every path to a next state. The input conditions required to traverse each path, along with the present state, form a product term as shown in Fig. 2. The next state code and output actions are assigned to PLA output pins. MACRO functions are treated as a single level called function which results in the assignment of a waiting state in the main line of flow and distinct states, as required, within

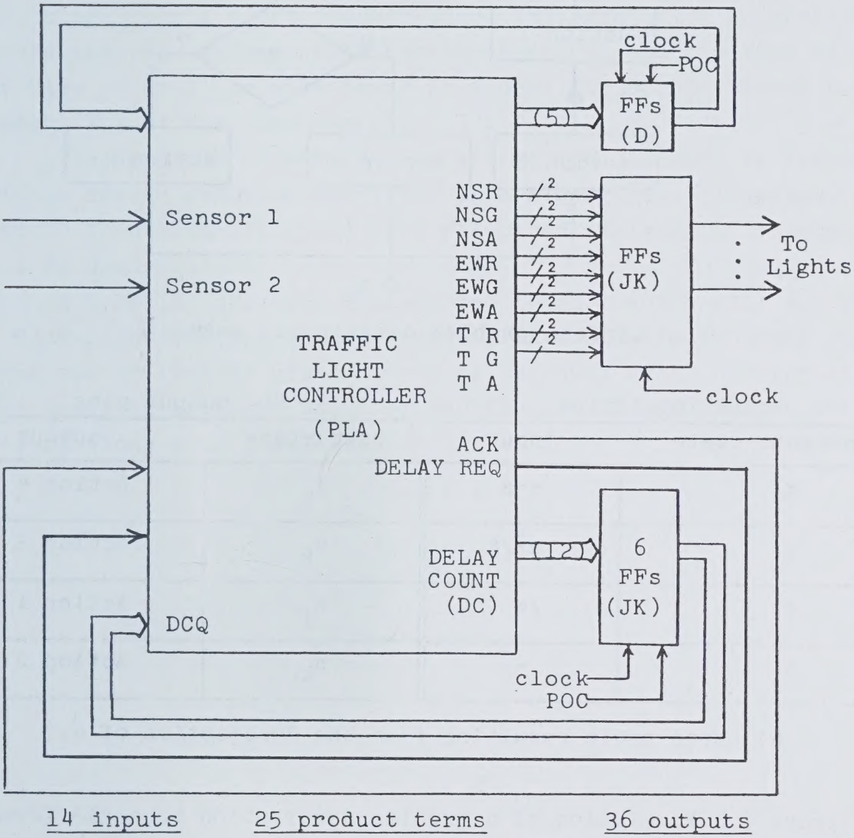
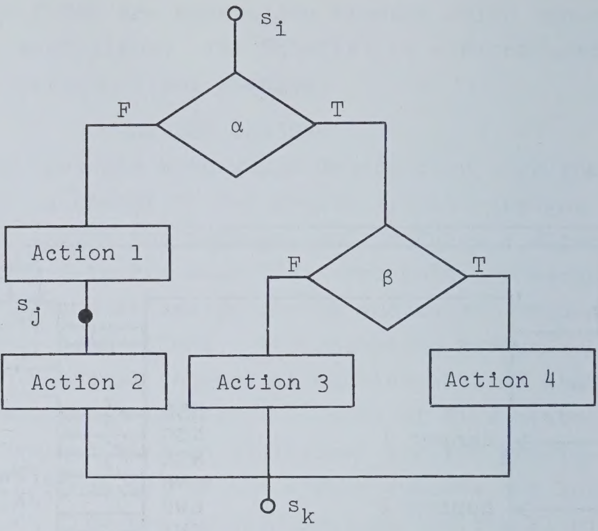

clock
POC
FFs (D)
(5)
NSR
NSG
NSA
EWR
EWG
EWA
T R
T G
T A
2
FFs (JK)
To Lights
clock
Sensor 1
Sensor 2
TRAFFIC LIGHT CONTROLLER (PLA)
ACK
DELAY REQ
DELAY COUNT (DC)
(12)
6 FFs (JK)
DCQ
clock
POC
14 inputs
25 product terms
36 outputs

Figure 1. Traffic controller hardware


s_i
F
α
T
Action 1
F
β
T
s_j
Action 2
Action 3
Action 4
s_k

a) a typical problem description segment

PLA input pins		PLA output pins	
present state	input	next state	output
s_i	$\alpha \cdot \beta$	s_k	Action 4
s_i	$\alpha \cdot /\beta$	s_k	Action 3
s_i	$/\alpha$	s_j	Action 1
s_j	--	s_k	Action 2

b) state table resulting from the description of a_1

Figure 2. Resolution of a problem description to a PLA form

the MACRO. The PLA code which results from the language description of example 1 is given in Table 1.

CONCLUSION

The use of language as the vehicle for system design has a number of advantages over methods requiring the generation and manipulation of detailed logic equations. The PLA has an advantage over a microcomputer-based implementation in problems where the task is basically control-oriented and the size of a problem is small or when there is a need for speeds beyond the capability of the microcomputer [3].

The approach presented here allows the designer to proceed with a design solution specified in a higher level language, and to then take advantage of a PLA-based implementation when one is desirable.

A compiler has been implemented in PL/1 which will accept a PL/M-like description and produce PLA configuration code for problems of limited size. There is, at this time, no provision for handling problems which exceed the I/O pin or product term restrictions of a single PLA.

Table 1. PLA code for traffic light controller

SENSOR 1	SENSOR 2	ACK	DCQ5	DCQ4	DCQ3	DCQ2	DCQ1	DCQ0	DELAY REQ	STATE VARS				DELAY REQ	STATE VARS				NSR	NSG	NSA	EWR	EWG	EWA	T R	T G	T A	DC5	DC4	DC3	DC2	DC1	DC0	ACK
										0	0	0	0	0	0	0	0	1	S			S			S									
										0	0	0	1	0	0	0	1	0	R	S								S			S	S	S	
		0								0	0	1	0	1	0	0	1	0																
		1								0	0	1	0	0	0	0	1	1		R	S										S			
		0								0	0	1	1	1	0	0	1	1																
		1								0	0	1	1	0	0	1	0	0	S		R													
										0	1	0	0	0	0	1	0	1				R	S						S	S				
		0								0	1	0	1	1	0	1	0	1																
		1								0	1	0	1	0	0	1	1	0					R	S							S			
		0								0	1	1	0	1	0	1	1	0																
		1								0	1	1	0	0	0	1	1	1				S		R										
0	0									0	1	1	1	0	0	0	0	1																
1										0	1	1	1	0	1	0	0	0							R	S				S	S	S		
	1									0	1	1	1	0	1	0	0	0							R	S				S	S	S		
		0								1	0	0	0	1	1	0	0	0																
		1								1	0	0	0	0	1	0	0	1								R	S				S			
		0								1	0	0	1	1	1	0	0	1																
		1								1	0	0	1	0	0	0	0	1							S		R							
			0	0	0	0	0	1	1	-	-	-	-																					1
								0	1	-	-	-	-																			T		
							0	0	1	-	-	-	-																		T			
						0	0	0	1	-	-	-	-																	T				
					0	0	0	0	1	-	-	-	-																T					
				0	0	0	0	0	1	-	-	-	-															T						
			-	-	-	-	-	-																									T	

References

[1] W.N. Carr and J.P. Mize, MOS/LSI Design and Application, McGraw-Hill, NY, 1972, pp. 229-258.

[2] J.C. Vermeulen, "Programmable Logic Arrays: An Alternative to Random LSI", IBM Tech. Report TR 44-0285, IBM GPD, Boulder, CO, April 20, 1976, pp. 1-29.

[3] J.C. Louge, N.F. Brickman, F. Howley, J.W. Jones and W.W. Wu, "Hardware Implementation of a small system in Programmable Logic Arrays", IBM J. Res. Develop., 19, March, 1975, pp. 110-119.

[4] J.W. Jones, "Array Logic Macros", IBM J. Res. Develop., 19, March, 1975, pp. 120-126.

[5] K. Gorman, "The Programmable Logic Array: A new approach to microprogramming", EDN, Nov. 1973, pp. 68-75.

[6] G. Reyling, "PLAs enhance digital processor speed and cut component count," Electronics, Aug. 1974, pp. 109-114.

[7] D. Gries, Compiler Construction for Digital Computers, Wiley, 1971.

DIGITAL SYSTEM DESIGN USING 4-VALUED LOGIC DEVICES

ANTHONY S. WOJCIK
Department of Computer Science
Illinois Institute of Technology
Chicago, Illinois 60616

ABSTRACT

With the recent announcements of the development of 4-valued I^2L circuits, digital system designers will soon have a new family of circuits at their disposal. This paper considers several aspects of system design using 4-valued logic elements. In particular, arithmetic and logic operations are considered. Further, some ramifications of "thinking" in two values rather than in four values are discussed, and some observations concerning Postian and Boolean 4-valued logics are made.

INTRODUCTION

With the advent of Signetics' 4-valued I^2L circuits [1-4], digital system designers are presented with the prospect of having to incorporate these circuits into system designs. Besides having to ascertain the electronic characteristics of these devices, the designer will be faced with the requirements of understanding their logical characteristics and of utilizing them in a predominantly binary world. However, it should be pointed out that Signetics claims that their circuit design technique is not limited to 4-valued logic. For practical considerations, such as the desire to use existing, standard binary memory and to interface with binary systems, 4-valued logic elements were initially developed. Hence, there is the prospect of the designer having several families of different multi-valued circuits to understand and to utilize. In this paper, only 4-valued logic elements are considered; but many of the observations that are made are relevant to any non-binary logic family that is based on the Signetics circuit design approach.

Given a family of 4-valued devices, the designer is faced with several considerations. Should the user of a system incorporating these devices continue to think binary, so that the 4-valued logic of the hardware must be transparent to him? Or, should the user think 4-valued? Further, will the non-binary devices be used to implement a system based on Post 4-valued logic or on Boolean 4-valued logic?

The purpose of this paper is to begin to address these issues and to consider several aspects of digital system design using 4-valued logic components.

BASIC PRINCIPLES

The logic elements developed by Signetics are based on the Post 4-valued logic system, P(4) [5]. The logic values in the system are denoted 0,1,2 and 3. The lattice for P(4) is shown in Figure 1a. Basically, the logic functions that are implemented are:

$$\left.\begin{array}{ll}\text{Maximum:} & \text{MAX}\ (X,Y) = X + Y \\ \text{Minimum:} & \text{MIN}\ (X,Y) = X \cdot Y\end{array}\right\}\ \text{see Table 1a and 1b}$$

Negation: $\overline{X} = 3 - X$ (sometimes called Complementation)

Multiplexor: $T(p,q,r,s;X) = p \cdot X^{(3000)} + q \cdot X^{(0300)} + r \cdot X^{(0030)} + s \cdot X^{(0003)}$

where $p,q,r,s,X \in \{0,1,2,3\}$

and

$$X^{(abcd)} = \begin{cases} a \text{ if } X = 3 \\ b \text{ if } X = 2 \\ c \text{ if } X = 1 \\ d \text{ if } X = 0 \end{cases} ,a,b,c,d \in \{0,1,2,3\}$$

The Multiplexor, or T-gate, can be written as $X^{(p\ q\ r\ s)}$. For notational purposes, the identity function $X^{(3210)}$ is written as X. These logic functions form a set of operations that is sufficient to implement any 4-valued logic function [3,5].

The Boolean 4-valued logic system, B(4), whose lattice is shown in Figure 1b, has as its basic set of logic functions:

$$\left.\begin{array}{r}\text{OR}\ (X,Y) = X + Y \\ \text{AND}\ (X,Y) = X \cdot Y\end{array}\right\}\ \text{See Table 1c and 1d}$$

$\overline{X} = 3 - X$ (correctly called Complementation)

It should be noted that B(4) is not complete in that not all 4-valued functions can be implemented [5].

Since binary information will be encoded into 4-valued data, binary 0 (1) will be associated with the 0 (3) of a 4-valued system. The reason for this is that 0 (1) is the zero (unit) of the 2-valued lattice, while 0 (3) is the zero (unit) of either P(4) or B(4). To simplify notation, we will write 0 and 3 for the two binary values.

DESIGN CONSIDERATIONS

Initially we consider design using the Signetics elements in such a way that the user will be assumed to be "thinking" in binary. The shortcomings of this assumption will be discussed.

Let us assume that we will use binary memory where, for illustrative purposes, each word consists of 8 bits, with B_0 representing the least significant bit of a word. The encoding process will take two consecutive bits, B_iB_{i-1}, of a binary word to form one quaternary digit (quat), Q_j, of the quaternary word. The least significant quat is represented by Q_0. The encoding (decoding) scheme is shown in Table 2. Using the Postian functions, we obtain the encoding and decoding equations:

Encoding: $Q_j = 1 \cdot B_{i-1} + 2 \cdot B_i + B_i \cdot B_{i-1}$

Decoding: $B_i = Q_j^{(3300)} = T(3,3,0,0; Q_j)$

$B_{i-1} = Q_j^{(3030)} = T(3,0,3,0; Q_j)$

We will assume that a quaternary operand is stored in a register, with each quat position of the register implemented by a 4-valued memory element [6]. Each memory element, Figure 2, has both negated and true outputs, is a clocked device, and can be cleared. Now let us consider a few selected arithmetic and logical operations that might be performed on the encoded data.

Addition/Subtraction: These operations are straightworward to implement. Logic equations for addition can be generated [7] and implemented with the Postian functions. A 4s complement number representation could be used in which the quaternary number will be ≥ 0 if the high order quat, Q_{n-1}, is ≤ 1, and will be < 0 if $Q_{n-1} \geq 2$. In This system sign detection is determined by:

$$Q_{n-1}^{(3300)} = \begin{cases} 3 \text{ if } Q_{n-1} = 2 \text{ or } 3 \text{ , negative} \\ 0 \text{ if } Q_{n-1} = 0 \text{ or } 1 \text{ , positive or zero.} \end{cases}$$

Letting C_n represent the carry out of the Q_{n-1} position, overflow and underflow conditions are given as:

Overflow: $C_n = 0$ and $Q_{n-1} = 2$ or 3

Underflow: $C_n = 1$ and $Q_{n-1} = 0$ or 1.

Since C_n can only be 0 or 1, we need only implement the function

$$T\ (0,0,0,3;\ C_n) = C_n^{(0003)}$$

and use it with the sign detection function to determine these conditions.

Note that the designer needs to expand his normal technique of checking if "bits" are on or off to checking what level a quat is at.

Shifting: The assumption that the user is thinking in binary makes the implementation of shift instructions quite complex. Since the binary information is encoded, merely shifting the quaternary operand does not correctly correspond to a single shift of the corresponding binary operand. Figure 3a and 3b illustrates the problem. Clearly, the problem lies in the fact that shifting one quat results in shifting two bits. So, the quaternary register cannot be made into a simple shift register to implement the required binary shifts.

One way to implement the binary shifts is to compute a Shift Function, SF, of the quaternary digits. Table 3 shows the value of a quat that is required to implement binary shifts to the left or right. It can be seen that the basic Shift Function is identical for left and right shifts, but that for a left shift, the Shift Function value becomes Q_i, and for a right shift, the value becomes Q_{i-1}. The SF equation is:

$$SF(Q_i,Q_{i-1}) = 1 \cdot Q_{i-1}^{(3300)} + 2 \cdot Q_i^{(3030)} + Q_i^{(3030)} \cdot Q_{i-1}^{(3300)}$$

Special conditions to be noted are:

Logical Left Shift: $SF(Q_0, Q_{n-1})$ generates value for Q_0

Logical Right Shift: $SF(Q_0,Q_{n-1})$ generates value for Q_{n-1}

Arithmetic Left Shift: $SF(Q_0,0)$ generates value for Q_0

Arithmetic Right Shift: $SF(0,Q_{n-1})$ generates value for Q_{n-1}.

Clearly, the implementation of binary shift operations in 4-valued hardware requires the addition of a significant amount of extra logic.

Logical AND / OR: Up until now, it was not necessary to emphasize the differences between B(4) and P(4). However, for logical operations, these differences are important. Since the user is thinking binary and requires the ANDing and ORing of operands, these logical operations have to be implemented using the functions which are based on P(4). Table 1c and 1d describe the Boolean OR and AND operations on B(4) that the user requires. The MAX and MIN operations on P(4), Table 1a and 1d, are not the same as the OR and AND operations. Hence, the required Postian implementations of the Boolean operations would be:

$$OR\ (X,Y) = X + Y + X^{(0030)} . Y^{(0300)} + X^{(0300)} . Y^{(0030)}$$

$$AND\ (X,Y) = X \cdot Y \cdot (X^{(3033)} + Y^{(3303)}) \cdot (X^{(3303)} + Y^{(3033)}).$$

Again, the constraint of thinking in binary significantly increases the complexity of the logic design.

OBSERVATIONS

The preceeding section attempts to point out some of the problems that arise when the designer is forced to use logic elements based on P(4) to implement a Boolean system that is required by a user thinking in binary. Granted, it appears that design problems can be overcome, but the question that arises is why should the user continue to "think" in binary? Clearly, users are already accustomed to thinking in octal and hexadecimal, not to mention decimal. Realistically, requiring him to think in a 4-valued system is not unreasonable. Further, learning a logic system based on Post algebras and manipulating 4-valued (or n-valued) data in such a system is not an insurmountable task.

From the perspective of the designers, implementing a binary system with P(4) logic is not only an artificial use of the logic components but also a waste of the potential power of a non-binary logic system. The designer should be attempting to discover appropriate uses of multi-valued systems and techniques for designing such systems. Some recent studies [6,8,9] have shown that the designer can make use of existing binary design techniques and readily extend these concepts to the multi-valued case.

It may be quite some time before multi-valued systems are seriously considered. However, the potential use of non-binary systems should not be hampered by the prevalent binary world. The designer and user of such systems should begin to become acclimated to the true multi-valued environment of Post algebra and Postian logic functions.

ACKNOWLEDGMENT

This research was supported by the National Science Foundation under Grant MCS 74-01409A01.

REFERENCES

[1] "Signetics readies first circuits using multi-level logic," Electronics, October 14, 1976, p.25.

[2] "Four-level logic coming next year from Signetics," Electronics, October 28, 1976, pp.31-32.

[3] T. Dao, K. Russell, D. Preedy, and E. J. McCluskey, "Multilevel I^2L with threshold gates," paper presented at the International Solid-State Circuits Conference, February 16-18, 1977, Philadelphia.

[4] T. Dao, "The year of multi-valued circuit realizations, "invited presentation at the Seventh International Symposium on Multiple-Valued Logic, May 24-27, 1977, Charlotte, N.C.

[5] A. S. Wojcik and G. Metze, "An analysis of some relationships between Post and Boolean algebras," J.ACM, Vol. 21, No. 4, pp. 680-696, October 1974.

[6] M. S. Wills, "A study of multi-valued logic R-flops," M.S. Thesis, Department of Computer Science, Illinois Institute of Technology, Chicago, May 1977.

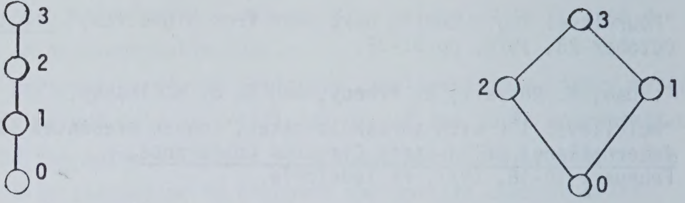
[7] A. S. Wojcik and G. Metze, "On the Cost of Base N Adders," 1EEE Transactions on Computers, Vol. C-20, No. 10, October 1971, pp. 1196-1203.

[8] S. J. Sheafor, "The design of multiple-valued asynchronous sequential circuits," Coordinated Science Laboratory Report R-663, University of Illinois at Urbana-Champaign, September 1974.

[9] A. S. Wojcik, "The Design of Multi-Valued Asynchronous Modules," Department of Computer Science Report No. 76-1, Illinois Institute of Technology, Chicago, November 1976.

[10] T. C. Yang, "The decomposition of multi-valued sequential machines," Department of Computer Science Report No. 77-12, Illinois Institute of Technology, Chicago, July 1977.

Figure 1: 4-valued lattices.



a) Lattice of P(4).

b) Lattice of B(4).

Table 1: 4-valued functions.

+	Y			
X	0	1	2	3
0	0	1	2	3
1	1	1	2	3
2	2	2	2	3
3	3	3	3	3

a) MAX(X,Y)

·	Y			
X	0	1	2	3
0	0	0	0	0
1	0	1	1	1
2	0	1	2	2
3	0	1	2	3

b) MIN(X,Y)

$\oplus$	Y			
X	0	1	2	3
0	0	1	2	3
1	1	1	3	3
2	2	3	2	3
3	3	3	3	3

c) OR(X,Y)

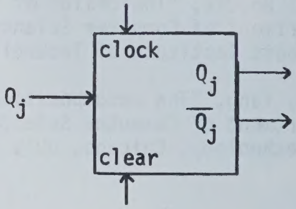
$\odot$	Y			
X	0	1	2	3
0	0	0	0	0
1	0	1	0	1
2	0	0	2	2
3	0	1	2	3

d) AND(X,Y)

Table 2: Encoding/decoding scheme.

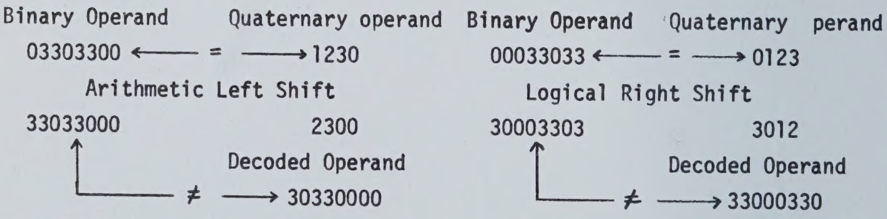
B_i	B_{i-1}	Q_j
0	0	0
0	3	1
3	0	2
3	3	3

Figure 2: 4-valued memory element.



clock
Q_j
Q_j
$\overline{Q_j}$
clear

Figure 3: Shift operations.



Binary Operand
Quaternary operand
03303300
=
1230
Arithmetic Left Shift
33033000
2300
Decoded Operand
≠
30330000
Binary Operand
Quaternary perand
00033033
=
0123
Logical Right Shift
30003303
3012
Decoded Operand
≠
33000330

a) Arithmetic shift.

b) Logical shift.

Table 3: Quat values for shifting.

	Q_{i-1}			
Q_i	0	1	2	3
0	0	0	1	1
1	2	2	3	3
2	0	0	1	1
3	2	2	3	3

Q_i

a) Left shift by 1.

	Q_{i-1}			
Q_i	0	1	2	3
0	0	0	1	1
1	2	2	3	3
2	0	0	1	1
3	2	2	3	3

Q_{i-1}

b) Right shift by 1.

AN ALGEBRA FOR THE REALIZATION OF SWITCHING FUNCTIONS USING A CERTAIN TYPE OF MOS PACKAGE

RAVINDRA NAIR and GERNOT METZE
Coordinated Science Laboratory
University of Illinois
Urbana, Illinois 61801

ABSTRACT

Switching functions may be realized using a certain type of MOSFET circuit package, similar to the Fairchild 3102 package. This paper outlines the conditions that must be satisfied by a switching function so that it may be realized minimally, in a series-parallel form. An algebra has been developed to obtain the smallest realizations whenever a minimal realization is possible, and near-minimal realizations in the other cases.

1. INTRODUCTION

This paper studies the realizability of switching functions using a MOSFET circuit package of the form shown in Fig.1. This type of package is commercially available, e.g., as the Fairchild 3102 integrated circuit where n=3.

The gates of the MOSFETs will be connected to the various input signals. The sources and drains of the MOSFETs may be interconnected in any manner with the restrictions that one of them must be connected to ground and that terminal z_0 must provide the output.

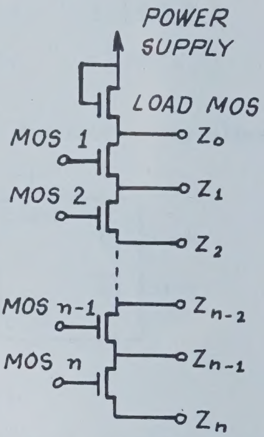


Fig. 1

Interconnections of the pins $z_0, z_1, \ldots, z_n$ in various ways make it possible to realize a host of switching functions. The limitations to realizing all switching functions of up to n variables arise from the fact that

a) the connection between gate i and gate i+1, for i=1,2,...,n-1 is inseparable, and

b) the output is restricted to be at pin z_0, i.e., at one end of the chain formed by the MOSFETs.

Examples of two different functions realized by the package, with n=3, are shown in the Fig.2.

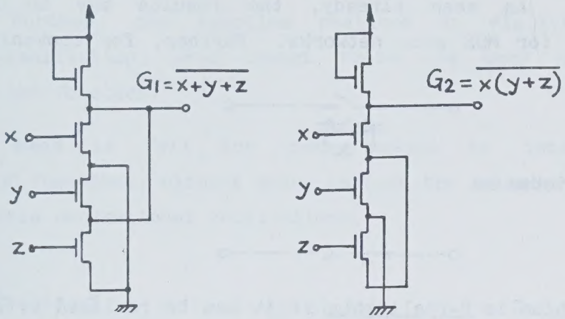


Fig. 2

The bilateral nature of the MOS gates makes them similar in their behavior to bilateral switches or relays. If each gate were replaced by a normally open relay contact, then the only difference would be that the output of the contact network would realize the complement of the output of the MOS gate network. We will now proceed to determine the kinds of functions that can be realized using this package.

2. DEFINITIONS

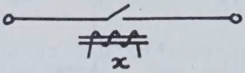
The basic postulates and theorems of switching algebra (e.g. [1]) will be assumed. Letter variables a,b,c,... will be used to designate MOS gates. Conjunctions will be denoted by ".", disjunction by "+", and complementation by "-".

A minimal form expression has the least number of literals among all equivalent expressions. A least form expression of n variables is a minimal form expression with exactly n literals, one for each variable. In a positive unate function, all variables are expressed in their true forms only.

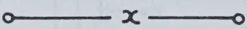
A network N is series-parallel if through each element (gate or relay) of N there is at least one path from the output terminal to the ground, not touching any junction twice, and no two of these paths pass through any element in opposite directions. (This definition has been modified from

[2].) A network which is not a series-parallel network is a bridge network.

In further discussion, bilateral relay switching networks will be considered. As seen already, the results may be applied equally effectively for MOS gate networks. Further, for convenience, the relay contact



will be depicted as



A function is P-realizable if it can be realized by a package of the form shown in Fig.1 with the gates replaced by normally open relay contacts. A function of k variables will be minimally P-realizable if it is P-realizable with no more than k elements in the chain, i.e. $n=k$. The function is assumed to be expressed in the least form.

The following observations can now be easily proved.

THEOREM 1: $\forall n>0$, $\exists k$, $0<k\leq n$, such that all functions of k or fewer variables are P-realizable with n elements in the package.

THEOREM 2: $\forall k>0$, $\exists n$, $n\geq k$, such that all functions of k or fewer variables are P-realizable with n elements in the package.

THEOREM 3: A minimally P-realizable function must be unate. (Unateness is only a necessary condition!)

COROLLARY 1: A minimally P-realizable function realized using only the true form of the variables must be positive unate.

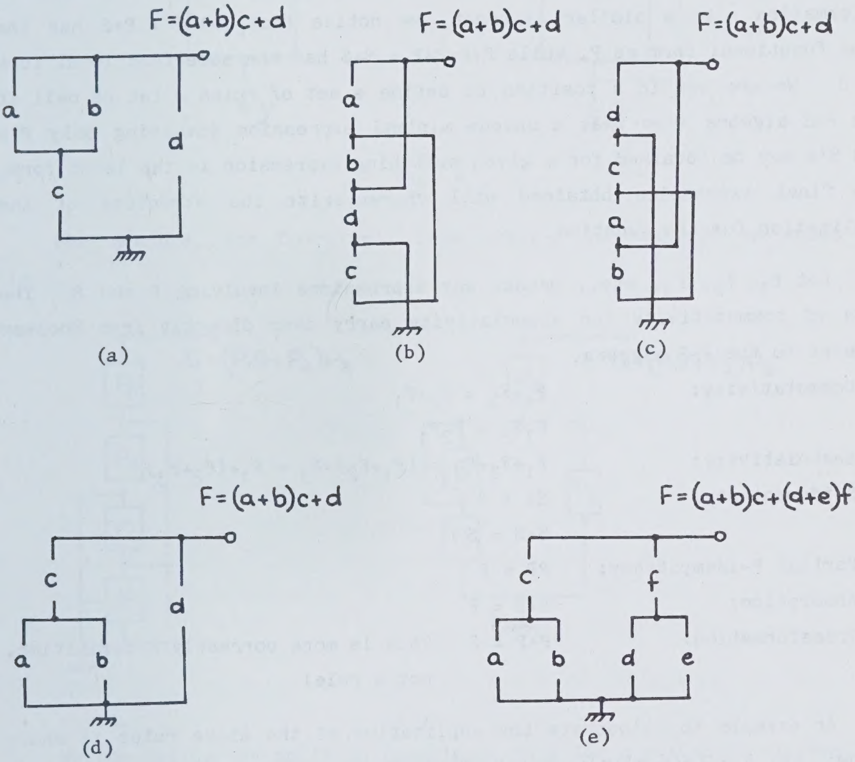
In future discussions, only positive unate functions will be considered. The extension to the entire class of unate functions is immediate, if both true and complemented forms are available for variables.

3. SERIES-PARALLEL CIRCUIT REALIZATION

A useful alternative definition for a series-parallel circuit is: A network is series-parallel if it is either a series or a parallel connection of two series-parallel networks. A single element (gate or relay) is also a series-parallel network.

The P-realizations shown in Figs.3(b) and 3(c) may be mapped to the conventional realization of Fig.3(a). However the conventional realization of Fig.3(d) which realizes the same function has no corresponding P-realization. Further, the function realized in Fig.3(e) has no corresponding P-realization, even though there are many conventional realizations for the function.

Hence the need is felt for some method to determine the P-realizability of functions, without going through the painstaking task of listing all possible conventional realizations.



(a) (b) (c)

(d) (e)

Fig. 3

We will attempt to classify switching functions according to the form that they assume in a P-realization. For instance, the expressions a, ab, abc will be assumed to have similar forms, distinct from the form assumed by a+b. This is because, for practical purposes, the behavior of the first three expressions in a more complex expression would be identical and would be distinguishable from the behavior of the last.

Since every function in its least form is characterized by the absence of repeating literals, we may proceed as follows.

We replace every literal in the switching expression by the letter P. Thus a gets transformed to P, ab to PP, abc to PPP and a+b to P+P. Further, P, PP and PPP represent an equivalent functional form that is distinct from P+P=S, say. Clearly, it would be convenient to replace PP and PPP by P, an equivalent functional form, without loss of structural information. By a similar argument, we notice that P+P+P = P+S has the same functional form as P, while P+P+P+P = S+S has the same functional form as S. We are now in a position to define a set of rules - let us call it the P-S algebra - so that a unique minimal expression involving only P's and S's may be obtained for a given switching expression in the least form. The final expression obtained will characterize the structure of the realization for the function.

Let F_1, F_2, F_3, etc., denote any expressions involving P and S. The laws of commutativity and associativity carry over directly from Boolean algebra to the P-S algebra.

1. Commutativity:	$F_1+F_2 = F_2+F_1$
	$F_1F_2 = F_2F_1$
2. Associativity:	$F_1+F_2+F_3 = (F_1+F_2)+F_3 = F_1+(F_2+F_3)$
3. S-idempotency:	SS = S
	S+S = S
4. Partial P-idempotency:	PP = P
5. Absorption:	P+S = P
6. Transformation:	P+P = S (This is more correctly a definition, not a rule)

An example to illustrate the application of the above rules is shown below: Let F = (a(b+c)+d)(e+f(g+h)+k)+l(m+n). If G denotes the equivalent functional form then

G = (P(P+P)+P)(P+P(P+P)+P)+P(P+P)
= (PS+P)(P+PS+P)+PS by Rule 6
= (PS+P)(P+P+PS)+PS by Rule 1
= (PS+P)((P+P)+PS)+PS by Rule 2
= (PS+P)(S+PS)+PS by Rule 6

No further simplification is possible.

We can now associate forms for P and S as shown in Fig.4, and observe that a P-realization is possible as long as there exists some numbering of the forms 1,2,3,...,k such that the terminal y of form i is connected to the terminal x of i+1, $1 \leq i < k$.

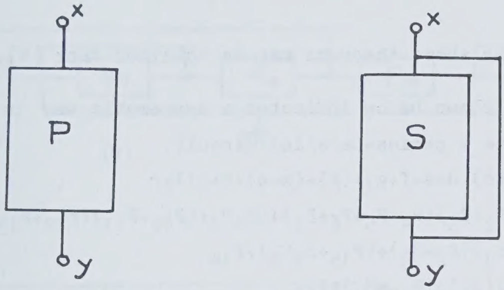


Fig. 4

For example, the functional from (PS+P)P is realized as shown in Fig.5.

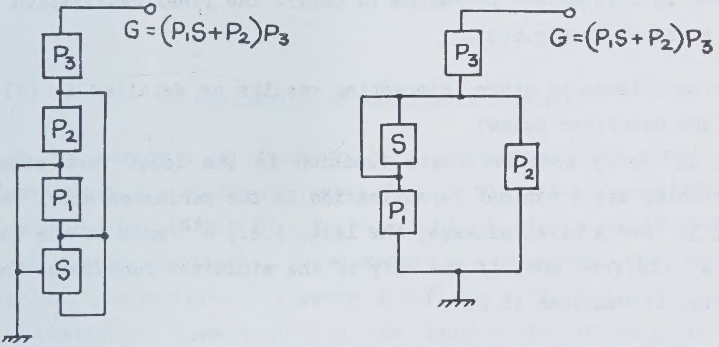


Fig. 5

We now define the modified functional form for a switching expression. If the minimized functional form for an expression is SG or PG then its modified functional form is G. It is important to note that the suggested modifications can be carried out only on the entire expression for the function to be realized. It cannot be used to reduce subexpressions within expressions.

THEOREM 4: If the modified functional form for a switching expression is P-realizable, then the original functional form, without modifications, is P-realizable.

THEOREM 5: A positive unate switching function expressed in the least form is minimally P-realizable if and only if there is no more than one occurrence of PS in the modified functional form for the corresponding expression.

Proofs of the above theorems may be obtained from [4].

The example shown below indicates a systematic way in which one would attempt to realize a series-parallel circuit.

Let $F = (a+b+c)(d+e+f+g)+(kl+(m+n)(o+p))+r$

Hence

$$G = (P_1+P_2+P_3)(P_4+P_5+P_6+P_7)+(P_8P_9+(P_{10}+P_{11})(P_{12}+P_{13}))+P_{14} \quad (1)$$

$$= (P_1+S_1)(S_2+S_3)+(P_{15}+S_4S_5)+P_{14} \quad (2)$$

$$= (P_{16})(S_6)+(P_{15}+S_7)+P_{14} \quad (3)$$

$$= P_{16}S_6+P_{17}+P_{14} \quad (4)$$

$$= P_{16}S_6+S_8 \text{ , which is indeed P-realizable} \quad (5)$$

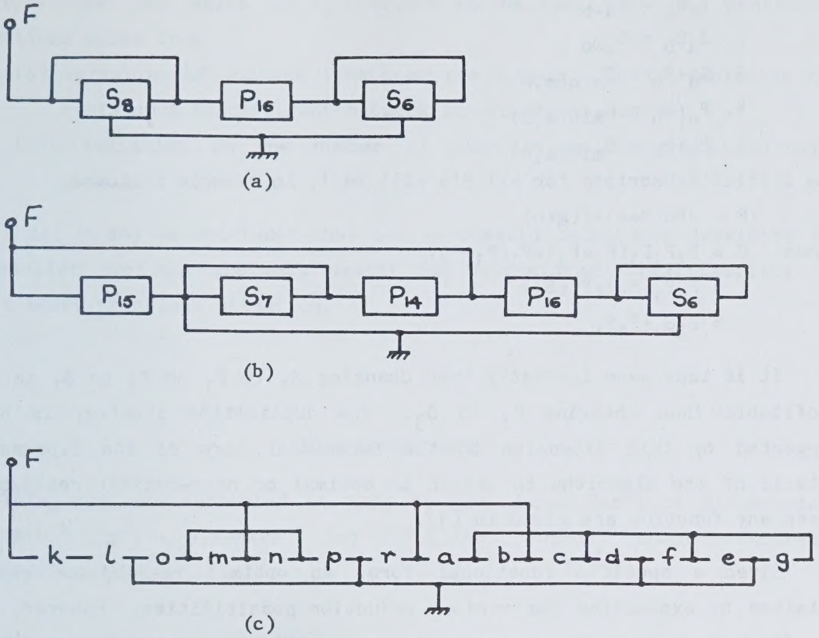
Fig.6(a) shows the P-realization for the relation (5). From this realization, the realization of Fig.6(b) is obtained for relation (3). We may proceed in this manner backwards to obtain the final realization of the function as shown in Fig.6(c).

Theorem 5 leads to other interesting results as detailed in [4], some of which are described below:

COROLLARY 2: Every positive unate function in the least form with 5 or fewer variables has a minimal P-realization in the series-parallel form.

COROLLARY 3: For a given package, the last, i.e., n^{th} gate in the chain is grounded at its free end, if and only if the minimized functional form for the function it realizes is P.

Finally, it must be said that for switching expressions which are not in the least form, similar techniques may be used to determine a P-realization in the series-parallel form. For instance, every duplicate occurrence of a variable may be considered a separate new variable. However, there is often the possibility of obtaining a minimal P-realization by using bridges. Such realizations have been studied in [4].



(a)

(b)

(c)

Fig. 6

4. CONVERSION TO REALIZABLE CIRCUITS

Let us consider a function F = abc(d+e)+f(g+h) which has the functional form G = PS+PS and hence is not minimally P-realizable. With the knowledge that (P+P)S+PS = SS+PS = S+PS is also a valid P-realizable form, we could duplicate 3 elements to transform the function to F = (abc+abc)(d+e)+f(g+h). However P(S+P)+PS = PP+PS = P+PS is also a valid P-realizable form requiring the duplication of only one element, instead of three, using the transformation F = abc(d+e+e)+f(g+h).

For the series-parallel case, the P-S algebra may be used to obtain the optimal realization for any function, in most cases, and near-optimal realizations in the others. Each P and S in the functional form is now subscripted by an integer which indicates the least number of elements that need to be duplicated to convert that S to a P or P to an S. When absorbing P's and S's, the subscripts will change according to the following rules:

1. $P_aP_b = P_{a+b}$
2. $S_aS_b = S_{a+b}$
3. $S_a+S_b = S_{\min(a,b)}$
4. $P_a+S_b = P_{\min(a,b)}$
5. $P_a+P_b = S_{\min(a,b)}$

The initial subscripts for all P's will be 1. An example follows:

$$F = abc(d+e)+f(g+h)$$

Hence
$$\begin{aligned} G &= P_1P_1P_1(P_1+P_1)+P_1(P_1+P_1) \\ &= P_2P_1(S_1)+P_1(S_1) \\ &= P_3S_1+P_1S_1 \end{aligned}$$

It is thus seen instantly that changing S_1 to P_1 or P_1 to S_1 is more profitable than changing P_3 to S_3. The duplication strategy is hence suggested by this extension to the functional form of the expression. Details of the algorithm to obtain an optimal or near-optimal realization given any function are given in [4].

Given a specific functional form, an optimal realization can be obtained by exhausting the various reduction possibilities. However, when the final functional form contains 3 or more PS's the number of cases to be examined becomes too large to merit the search for optimality. In any case, the algorithm just described does lead to the optimal solution in most cases.

5. DISCUSSION

The central idea of this paper is to indicate systematic techniques to realize functions using a specific organization described in Section 1. While it is difficult to determine exactly what fraction of all possible functions can be realized using the package minimally, it is clear from the results obtained here and those obtained for bridge circuits that a large majority of the functions with few (around 10) variables can indeed be realized using the package.

When the number of variables becomes large, the procedures for giving a minimal P-realization or for obtaining an optimal transformation to a P-realizable form become quite complex. In any case, implementation difficulties will generally forbid the chaining of a large number of identical MOSFET gates, so that there is a limit placed on the number of variables that may exist in a function to be realized. The practical limitations arise from:

(a) variation of voltage levels at the output, due to variation of resistance in different network interconnections, and

(b) limitation on the number of pins in an integrated circuit package.

Thus, it may be concluded that the systematic techniques developed in the earlier sections are very useful for determining P-realizability in almost every practical situation.

ACKNOWLEDGEMENT

This work was supported in part by the Joint Services Electronics Program(U.S.Army, U.S.Navy and U.S.Air Force) under Contract DAAB-07-72-C-0259.

REFERENCES

[1] Kohavi,Z., Switching and Finite Automata Theory, McGraw-Hill, New York, 1970.

[2] Riordan,J. and Shannon,C.E., "The number of two-terminal series-parallel networks," J. Math. and Phys., Vol.21, pp.88-93, Aug.1942.

[3] Miller,R.E., "Formal Analysis and Synthesis of Bilateral Switching Networks," IRE Trans. Elec. Comput., Vol.EC-7, pp.231-244, Sept.1958.

[4] Nair,R. and Metze,G., "An Algebra for the Realization of Switching Functions using a certain type of MOSFET package," Coordinated Science Laboratory Report R-775, University of Illinois, Urbana, July 1977.

A MULTICODE SINGLE TRANSITION TIME STATE ASSIGNMENT FOR ASYNCHRONOUS SEQUENTIAL MACHINES*

JON G. KUHL and SUDHAKAR M. REDDY
Division of Information Engineering
The University of Iowa
Iowa City, Iowa 52242

ABSTRACT

A multicode single transition time state assignment for normal mode asynchronous sequential machines is given. The proposed state assignment requires $2\lceil \log_2 n \rceil$ state variables for flow tables with n states.

I. INTRODUCTION

In this paper we give a procedure to derive multicode STT state assignments that require $2\lceil \log_2 n \rceil$ state variables. In most cases the state assignments being proposed significantly reduce the required number of state variables as compared to earlier known unicode STT assignments [2,4-6]. This comparison, for several randomly generated flow tables, is made in Table 1. The data for Table 1 is from reference 5. Equally important is the fact that the design equations required for the asynchronous sequential circuits using the proposed state assignments can be generated directly, and by a simple procedure, from the flow tables.

II. PRELIMINARIES

In this section the assumptions being made, the notation to be used and definition of several terms are given. The following assumptions are made on the asynchronous sequential machines (for which the state assignments are to be made) and the asynchronous sequential circuits realizing them.

TABLE 1. Comparison of several STT state assignment techniques for some example flow tables

Table Ref. No. (Ref. 5, Table 1)	No. of rows	No. of columns	No. of K-sets	Number of State Variables Required			
				Ref. [5]	Ref. [6]	Ref. [4]	Proposed Assignment
38	7	3	11	4	11	9	≤ 6
6	6	4	12	4	12	9	≤ 6
32	9	4	16	6	16	16	≤ 8
24	15	10	46	13	46	16	≤ 8
28	25	10	72	27	72	25	≤ 10
22	39	10	106	37	106	36	≤ 12
23	36	11	115	43	115	36	≤ 12

---

* The research reported has been supported by National Science Foundation Grant ENG72-04142 A02.

(a) The asynchronous sequential machines are represented by normal mode flow tables [10] (for normal mode flow tables the next state entries are equal to the next stable states or, equivalently, each unstable state leads directly to a stable state).

(b) The asynchronous circuit realizing the given normal mode flow table is to operate in fundamental mode, i.e. the inputs to the circuit are not changed until it is in a stable state.

(c) The delays in the feedback paths are adequate to eliminate hazards [1].

Assumptions (a) through (c) have to be satisfied for the general application of most of the known results on asynchronous state assignments [1-7]. Elimination of inserted delays is possible under certain conditions [8,9].

Next we give most of the notation used in this paper.

Notation:

M is the flow table of a given asynchronous sequential machine.

N_M is the asynchronous sequential circuit to be designed to realize M.

S is the set of states of M.

I is the set of inputs of M and N_M.

O is the set of outputs of M and N_M.

S_N is the set of states of N_M (note that members of S_N are binary vectors); we will use primed symbols to denote members of S_N, for example $s_1' \varepsilon S_N$.

n is the number of states of M.

m is the number of binary input variables of M; i.e., I is a subset of the set of 2^m binary m-tuples.

ℓ is the number of binary output variables of M.

$r = \lceil \log_2 n \rceil$, where $\lceil x \rceil$ is the least integer greater than or equal to x.

B^a is the set of 2^a binary a-tuples.

y_i is the ith current state variable of N_M.

Y_i is the ith next state variable of N_M.

$i_j \rightarrow i_k$ is an input transition or change from $i_j \varepsilon I$ to $i_k \varepsilon I$.

$\delta : S \times I \rightarrow S$ is the next state map of M.

$\delta_N : S_N \times I \rightarrow S_N$ is the next state map of N_M.

$\lambda : S \times I \rightarrow 0$ is the output map of M.

$\lambda_N : S_N \times I \rightarrow 0$ is the output map of N_M.

$\circ$ is the concatenation operator for binary strings, e.g. $01 \circ 10 = 0110$.

Definition 1: A state $s_i \varepsilon S$ is said to be a *stable state* under input $i_j \varepsilon I$ if and only if $\delta(s_i, i_j) = s_i$.

Definition 2: Total states of M are the ordered pairs (s_i, i_j), $s_i \varepsilon S$ and $i_j \varepsilon I$.

Definition 3: A total state (s_i, i_j) of M is said to be a *stable total state under* i_j if and only if s_i is a stable state under input i_j.

Although the preceding definitions introduced several terms for M, similar terms can be defined for N_M also, by interchanging δ and δ_N, and M and N_M. In the sequal we will use these terms for N_M also, without defining.

Since normally there are restrictions on the input transitions for asynchronous sequential circuits (e.g., only single input variable changes are allowed for normal fundamental mode operation [1]), we next define admissible input transitions.

Definition 4; An input transition $i_j \rightarrow i_k$ is called *an admissible input transition* if and only if input change from i_j to i_k is allowed for N_M.

Definition 5: *The parity of an input* $i_j \varepsilon I$, *written* $P(i_j)$, is even if the number of ones in i_j is an even integer and the parity is odd if the number of ones is an odd integer. For the sake of brevity we say that $P(i_j) = 0$ if $P(i_j)$ is even and $P(i_j) = 1$ if $P(i_j)$ is odd.

Definition 6: An input $i_j \varepsilon I$ is said to be *adjacent* to input $i_k \varepsilon I$ if and only if i_j and i_k differ in exactly one corresponding position.

Definition 7: An input $i_j \varepsilon I$ is called a *trivial input* if and only if there is only one stable state under i_j.

Definition 8: A stable state $s_i \varepsilon S$ under $i_j \varepsilon I$ is called an *inaccessible stable state* if and only if there does not exist a stable state $s_t \varepsilon S$ under input $i_k \varepsilon I$, $i_k \neq i_j$, such that $i_k \rightarrow i_j$ is an admissible input transition and $\delta(s_t, i_j) = s_i$. If s_i is not an inaccessible stable state then it is called an *accessible stable state*.

To illustrate Definitions 5 through 8 consider the flow table given in Table 2. Input 10 is a trivial input and stable state 2 under input 00 is an inaccessible stable state. P(00) = P(11) = 0, and P(01) = P(10) = 1. Input 00 is adjacent to inputs 01 and 10.

TABLE 2 Flow table of an example asynchronous sequential machine

Inputs x_1 x_2

	00	01	11	10
1	①, 0	①, 0	2	3
2	②, 1	3	②, 1	3
3	③, 1	③, 1	4	③, 1
4	1	④, 0	④, 0	3

Next state, output z_1 (circled entries are the stable states)

III. STATE ASSIGNMENT FOR SINGLE INPUT VARIABLE CHANGE FLOW TABLES

In this section, in addition to the assumptions (a) through (c) made in the last section, we assume that (d) only single input variables change at a time. This implies that $i_j \to i_k$ is an admissible input transition if and only if either $i_j = i_k$ or i_j is adjacent to i_k.

Theorem 1, given next, forms the basis for the validity of the state assignment presented in this section.

Theorem 1: If only one input variable changes in an input transition, say $i_j \to i_k$, $i_j \neq i_k$, then $P(i_k) = \bar{P}(i_j)$, where $\bar{x}$ is the complement of x.

For the asynchronous sequential machines that admit assumptions (a) through (d), Theorem 1 essentially divides the set of inputs I into two disjoint subsets. Elements in one set have parity 0 and the other set has parity one, and all admissible input transitions, $i_j \to i_k$, $i_j \neq i_k$, occur from elements of one subset into elements of the other subset. Throughout the rest of this section we will assume that all input transitions considered in the sequel are admissible input transitions.

As indicated in the introduction, the state assignment to be proposed is a multicode STT state assignment that requires 2r state variables, where $r = \lceil \log_2 n \rceil$. The proposed state assignment assigns one or more stable states of S_N to each stable state of S. Since the determination of stable state requires the knowledge of input, we adopt the following notation.

Let $s_i \varepsilon S$ be a stable state under input $i_j \varepsilon I$. Then $A_{i_j}(s_i)$ is the subset of states of N_M that are stable under input i_j and are assigned to s_i by the state assignment being proposed.

For given s_i, $A_{i_j}(s_i)$ is to be determined by one of the Rules 1 through 3 given below.

Let $A_E : S \to B^r$ and $A_0 : S \to B^r$ be two arbitrarily chosen single valued functions, and let $s_i \varepsilon S$ be a stable state under input i_j (recall that $r = \lceil \log_2 n \rceil$, where n is the number of states of M). The next three rules specify $A_{i_j}(s_i)$.

Rule 1: This rule is applicable if and only if s_i is an inaccessible stable state or if i_j is a trivial input.

$$A_{i_j}(s_i) = A_E(s_i) \circ A_0(s_i)$$

Rule 2: This rule is applicable if and only if Rule 1 is not applicable and $P(i_j)$ is even.

$$A_{i_j}(s_i) = \begin{cases} \left(A_E(s_i) \circ A_0(s_p)\right) : \left(A_E(s_i) \circ A_0(s_p)\right) \varepsilon S_N \text{ and } s_p \varepsilon S, \text{ where } s_p \\ \text{is a stable state under input } i_k \varepsilon I,\ i_k \neq i_j, \text{ such that} \\ \delta(s_p, i_j) = s_i \text{ and } i_k \to i_j \text{ is an admissible input transition.} \end{cases}$$

Rule 3: This rule is applicable if and only if Rule 1 is not applicable and $P(i_j)$ is odd.

$$A_{i_j}(s_i) = \begin{cases} \left(A_E(s_p) \circ A_0(s_i)\right) : \left(A_E(s_p) \circ A_0(s_i)\right) \varepsilon S_N \text{ and } s_p \varepsilon S, \text{ where } s_p \\ \text{is a stable state under input } i_k \varepsilon I,\ i_k \neq i_j, \text{ such that} \\ \delta(s_p, i_j) = s_i \text{ and } i_k \to i_j \text{ is an admissible input transition.} \end{cases}$$

Since for certain stable states in M, more than one stable state in N_M is assigned, we need to specify the next state variable values and the values of outputs for these stable states of N_M to complete the STT assignment being proposed. Note that for STT assignments the next state variables need only be specified for those stable states of N_M that are assigned to some state of M, since the next state variable values for other states of N_M either are do not cares (denoted by $\emptyset$ in the sequel) or are implied by the requirement for STT assignments to allow all state variable values, that are required to change, to change in any order. The next state variable values for the stable states of N_M, identified in Rules 1 through 3, should be assigned by one of the rules among Rules 4 through 6 given below. First we need the following notation.

Notation:

If $s'_i \varepsilon S_N$, let s'_{iL} and s'_{iR} be the leftmost and the rightmost r bits of s'_i, respectively. This notation implies that

$$s'_i = s'_{iL} \circ s'_{iR} , \qquad \text{where } s'_{iL} \text{ and } s'_{iR} \varepsilon B^r .$$

Let $s'_i = s'_{iL} \circ s'_{iR}$ be a stable state of N_M under input i_j and $s'_i \varepsilon A_{i_j}(s_p)$, $s_p \varepsilon S$. Furthermore, let $i_j \rightarrow i_k$, $i_j \neq i_k$, be an admissible input transition. The next three rules specify $\delta_N(s'_i, i_k)$.

Rule 4: This rule applies if and only if $P(i_j)$ is even (and hence $P(i_k)$ is odd) and i_k is not a trivial input.

$$\delta_N(s'_i, i_k) = \emptyset \qquad \text{if} \qquad \delta(s_p, i_k) = \emptyset$$

and

$$\delta_N(s'_i, i_k) = s'_{iL} \circ A_0(s_q) \qquad \text{if} \qquad \delta(s_p, i_k) = s_q$$

Rule 5: This rule applies if and only if $P(i_j)$ is odd (and hence $P(i_k)$ is even) and i_k is not a trivial input.

$$\delta_N(s'_i, i_k) = \emptyset \qquad \text{if} \qquad \delta(s_p, i_k) = \emptyset$$

and

$$\delta_N(s'_i, i_k) = A_E(s_t) \circ s'_{iR} \qquad \text{if} \qquad \delta(s_p, i_k) = s_t$$

Rule 6: This rule is applicable if and only if i_k is a trivial input. Let $s_u \varepsilon S$ be the single stable state under input $i_k \varepsilon I$. Then,

$$\delta_N(s'_i, i_k) = \emptyset \qquad \text{if} \qquad \delta(s_p, i_k) = \emptyset$$

and

$$\delta_N(s_i, i_k) = A_E(s_u) \circ A_0(s_u) \qquad \text{if} \qquad \delta(s_p, i_k) = s_u$$

The outputs associated with the stable states of N_M are to be assigned by Rule 7 given below.

Rule 7: Let $s'_i \varepsilon A_{i_j}(s_k)$ be a stable state under $i_j \varepsilon I$. Then,

$$\lambda_N(s'_i, i_j) = \lambda(s_k, i_j) .$$

NOTE: By cross checking with Rules 1 through 3 it can readily be ascertained that if $\delta_N(s'_i, i_k) = s'_j$ then s'_j is a stable state under i_k.

Before we prove that the state assignment given by Rules 1 through 6 is a multicode STT assignment we will apply it to the flow table of Table 2 in Example 1, given below.

Example 1: Consider the flow table given in Table 2. For this flow table $r = 2$. State 2 is an inaccessible stable state under input 00, and 10 is a trivial input. Let $A_E : S \to B^2$ and $A_0 : S \to B^2$ be such that $A_E(1) = 00$, $A_E(2) = 01$, $A_E(3) = 10$, $A_E(4) = 11$, $A_0(1) = 00$, $A_0(2) = 01$, $A_0(3) = 10$ and $A_0(4) = 11$. Then,

$$\left.\begin{aligned}
A_{00}(1) &= (0000),\ (0011)\\
A_{00}(2) &= (0101)\\
A_{00}(3) &= (1010)\\
A_{01}(1) &= (0000)\\
A_{01}(3) &= (1010),\ (0110)\\
A_{01}(4) &= (1111)\\
A_{11}(2) &= (0100)\\
A_{11}(4) &= (1110),\ (1111)\\
A_{10}(3) &= 1010)
\end{aligned}\right\}\qquad \text{Equation 1}$$

The partial transition table showing the stable states given in Equation 1, the next state entries computed by applying Rules 4 through 6, and the outputs computed by Rule 7, is given in Table 3. By further requiring that the proposed state assignment be a single transition time state assignment we get the transition table of Table 4 for the N_M which realizes M given in Table 2. Note that the next state entries for some states of N_M, other than the stable states idetified in Equation 1, are specified by the requirements of STT state assignments (these are the entries with '*').

It is useful to look at the proposed state assignment intuitively. For example if $i_j \to i_k$ is an admissible input transition, and i_k is not a trivial input, then the values of the leftmost r state variable values (rightmost r state variable values) do not change if the parity of the input changes from even to odd (odd to even). Or in othe words the leftmost r state variables are the controlling variables for state transitions that result due to changes in input parity from even to odd, and the rightmost r state variables are the controlling variables for state transitions that result due to changes in input parity from odd to even.

We next prove, in Theorem 2, that the state assignment given by Rules 1 through 6 constitutes a valid multicode STT state assignment. We will need the following three definitions and Lemma 1 to state Theorem 2.

Definition 9: Let s'_i, $s'_j \varepsilon S_N$ and let $\{s'_i\ ;\ s'_j\}$ be the subset of states of S_N that are obtained by arbitrarily specifying the values of the state variables in the positions in which s'_i and s'_j differ. $\{s'_i\ ;\ s'_j\}$ is called a *transition path* of N_M.

Definition 10: A transition path $\{s'_i\ ;\ s'_j\}$ of N_M is called an *admissible transition path of N_M under input i_v* if and only if the following conditions are satisfied:

(i) there exists a state $s_p \varepsilon S$ that is stable under an input i_u and $s'_i \varepsilon A_{i_u}(s_p)$, and

Table 3 Partial transition table obtained by applying Rules 1 through 7 to the flow table of Table 2.

	Inputs x_1, x_2			
	00	01	11	10
0000	0000, 0	0000, 0	0100	1010
0001				
0010				
0011	0011, 0	0000		1010
0100		0110	0100, 1	1010
0101	0101, 1	0110		1010
0110	1010	0110, 1	1110	
0111				
1000				
1001				
1010	1010, 1	1010, 1	1110	1010, 1
1011				
1100				
1101				
1110		1111	1110, 0	1010
1111	0011	1111, 0	1111, 0	1010

Next state variable values, output (circled entries are the stable states).

Table 4 Transition table obtained by applying Rules 1 through 7 to the flow table of Table 2 and further requiring that the state assignment be an STT state assignment.

	Inputs x_1, x_2			
	00	01	11	10
0000	0000, 0	0000, 0	0100	1010
0001	Ø	0000*	Ø	1010*
0010	1010*	0000*	Ø	1010*
0011	0011, 0	0000	Ø	1010
0100	Ø	0110	0100, 1	1010
0101	0101, 1	0110	Ø	1010
0110	1010	0110, 1	1110	1010*
0111	0011*	0110*	Ø	1010*
1000	Ø	Ø	Ø	1010*
1001	Ø	Ø	Ø	1010*
1010	1010, 1	1010, 1	1110	1010, 1
1011	0011*	Ø	Ø	1010*
1100	Ø	Ø	Ø	1010*
1101	Ø	Ø	Ø	1010*
1110	1010*	1111	1110, 0	1010
1111	0011	1111, 0	1111, 0	1010

(ii) there exists an input $i_v \varepsilon I$ such that $i_u \to i_v$ is an admissible input transition and $\delta_N(s'_i, i_v) = s'_j$.

Lemma 1: Let $\{s'_i ; s'_j\}$ and $\{s'_u ; s'_v\}$ be two admissible transition paths of N_M under input $i_k \varepsilon I$ and let $s'_j \varepsilon A_{i_k}(s_p)$, $s'_v \varepsilon A_{i_k}(s_q)$, s_p, $s_q \varepsilon S$ and $s_p \neq s_q$.

Then the state assignment given by Rules 1 through 6 is an *STT state assignment* if and only if for all possible choices of s'_i, s'_j, s'_u, s'_v, s_p, s_q and i_k the two admissible input transition paths $\{s'_i ; s'_j\}$ and $\{s'_u ; s'_v\}$ do not have any element in common.

Theorem 2: The state assignment given by Rules 1 through 6 is an STT state assignment.

Design Equations:

Next we give an algorithm to write down design equations for the next state variables and the outputs of N_M. First we need the following notation.

Notation:

Let $y_{e1}, y_{e2}, \ldots, y_{er}$ be the leftmost r state variables, $y_{o1}, y_{o2}, \ldots, y_{or}$ the rightmost r state variables, $z_1, z_2, \ldots, z_\ell$ the output variables, and $x_1, x_2, \ldots, x_m$ the input variables of N_M. If $s'_i = A_E(s_t) \circ A_0(s_u)$ is a stable state of N_M, then let

$$\text{AND}\left(A_E(s_t)\right) \text{ be the product term } y^*_{e1} \cdot y^*_{e2} \cdot \ldots \cdot y^*_{er}$$

and

$$\text{AND}\left(A_0(s_u)\right) \text{ be the product term } y^*_{o1} \cdot y^*_{o2} \cdot \ldots \cdot y^*_{or} ,$$

where $y^*_{ei} = y_{ei}$ if the ith component of $A_E(s_t) = 1$, $y^*_{ei} = \bar{y}_{ei}$ if the ith component of $A_E(s_t) = 0$ and $y^*_{oi} = y_{oi}$ or $\bar{y}_{oi}$ depending on whether the ith component of $A_0(s_u)$ is 1 or 0, respectively, $1 \leq i \leq r$. If $i_k \varepsilon I$ then let $\text{AND}(i_k)$ be the product term $x^*_1 \cdot x^*_2 \cdot \ldots \cdot x^*_m$, where $x^*_j = x_j$ or $\bar{x}_j$ depending on whether the jth component of i_k is 1 or 0, respectively. Let I_E and I_0 be the subsets of I such that all inputs in I_E have even parity and all inputs in I_0 have odd parity.

Next we give Algorithm A to write design equations for N_M. The design equations to be given will be in sum of products form. These sums are not guaranteed to be irredundant, but have the advantage of not requiring the generation of the transition table and can be written directly from the flow table.

Algorithm A;

1. As described earlier, arbitrarily pick two maps $A_E : S \to B^r$ and $A_0 : S \to B^r$. Let $s_i \varepsilon S$ be a stable state under input $i_k \varepsilon I$.
2. If $s_i \varepsilon S$ is an inaccessible stable state and $i_k \varepsilon I_E$, then include the product of $\left(\text{AND}(A_0(s_i))\right)$ and $\text{AND}(i_k)$ in the sum for Y_{ej}, Y_{oj} and z_p, if $A_E(s_i)$ has a one in the jth position, $A_0(s_i)$ has a one in the jth

position and z_p is 1 in $\lambda(s_i, i_k)$, respectively, $1 \leq j \leq r$, $1 \leq p \leq \ell$.

3. If $s_i \varepsilon S$ is an inaccessible stable state, and $i_k \varepsilon I_0$, then include the product of $AND(A_E(s_i))$ and $AND(i_k)$ in the sum for Y_{ej}, Y_{oj}, and z_p, if the jth position of $A_E(s_i)$ is a 1, the jth position of $A_0(s_i)$ is a 1, and $z_p = 1$ in $\lambda(s_i, i_k)$, respectively, $1 \leq j \leq r$, $1 \leq p \leq \ell$.
4. If $i_k \varepsilon I$ is a trivial input and s_i is the single stable state under i_k, then include $AND(i_k)$ in Y_{ej}, Y_{oj} and z_p, if the jth position of $A_E(s_i)$ is 1, the jth position of $A_0(s_i)$ is a 1, and $z_p = 1$ in $\lambda(s_i, i_k)$, respectively, $1 \leq j \leq r$, $1 \leq p \leq \ell$.
5. If $s_i \varepsilon S$ is an accessible stable state, i_k is a non-trivial input in I_E, s_t is a stable state under input $i_j \varepsilon I_0$, $i_j \rightarrow i_k$ is an admissible input transition and $\delta(s_t, i_k) = s_i$ then include the product of $AND(A_0(s_t))$ and $AND(i_k)$ in the sums for Y_{ej}, Y_{oj}, and z_p, if the jth position in $A_E(s_i)$ is a one, the jth position of $A_0(s_t)$ is a one, and $z_p = 1$ in $\lambda(s_i, i_k)$, respectively, $1 \leq j \leq r$, $1 \leq p \leq \ell$.
6. If $s_i \varepsilon S$ is an accessible stable state, i_k is a non-trivial input in I_0, $s_t \varepsilon S$ is a stable state under input $i_j \varepsilon I_E$, $\delta(s_t, i_k) = s_i$ and $i_j \rightarrow i_k$ is an admissible input transition, then include the product term $AND(A_E(s_t))$ and $AND(i_k)$ in the sums for Y_{ej}, Y_{oj} and z_p, if the jth position in $A_E(s_t)$ is a one, the jth position in $A_0(s_i)$ is a one, and $z_p = 1$ in $\lambda(s_i, i_k)$ respectively, $1 \leq j \leq r$, $1 \leq p \leq \ell$.

Example 2: Consider the flow table given in Table 2 and let A_E and A_0 be the maps defined in Example 1. Design equations for N_M obtained by applying Algorithm A to this flow table are given below (in manually writing the design equations we have found it convenient to work with one stable total state at a time, determine the transitions into it and then use the appropriate step of Algorithm A).

$$Y_{e1} = y_{o1}\bar{y}_{o2}\bar{x}_1\bar{x}_2 + y_{e1}\bar{y}_{e2}\bar{x}_1x_2 + y_{e1}y_{e2}\bar{x}_1x_2 + y_{o1}\bar{y}_{o2}x_1x_2 + y_{o1}y_{o2}x_1x_2 + x_1\bar{x}_2$$

$$Y_{e2} = \bar{y}_{o1}y_{o2}\bar{x}_1\bar{x}_2 + \bar{y}_{e1}y_{e2}\bar{x}_1x_2 + y_{e1}y_{e2}\bar{x}_1x_2 + \bar{y}_{o1}\bar{y}_{o2}x_1x_2 + y_{o1}\bar{y}_{o2}x_1x_2$$
$$+ y_{o1}y_{o2}x_1x_2$$

$$Y_{o1} = y_{o1}y_{o2}\bar{x}_1\bar{x}_2 + y_{o1}\bar{y}_{o2}\bar{x}_1\bar{x}_2 + \bar{y}_{e1}y_{e2}\bar{x}_1x_2 + y_{e1}\bar{y}_{e2}\bar{x}_1x_2 + y_{e1}y_{e2}\bar{x}_1x_2$$
$$+ y_{o1}\bar{y}_{o2}x_1x_2 + y_{o1}\bar{y}_{o2}x_1x_2 + y_{o1}y_{o2}x_1x_2 + x_1\bar{x}_2$$

$$Y_{o2} = \bar{y}_{o1}y_{o2}\bar{x}_1\bar{x}_2 + y_{o1}y_{o2}\bar{x}_1\bar{x}_2 + y_{e1}y_{e2}\bar{x}_1x_2 + y_{o1}y_{o2}x_1x_2$$

$$z_1 = \bar{y}_{o1}y_{o2}\bar{x}_1\bar{x}_2 + y_{o1}\bar{y}_{o2}\bar{x}_1\bar{x}_2 + \bar{y}_{e1}y_{e2}\bar{x}_1x_2 + y_{e1}\bar{y}_{e2}\bar{x}_1x_2 + \bar{y}_{o1}\bar{y}_{o2}x_1x_2 + x_1\bar{x}_2$$

If gates are shared, ten AND gates and five OR gates (with a total of 68 gate inputs) are required to realize the equations above. An interesting property of the design equations derived by Algorithm A is that all product terms required in the sums for outputs, z_i's, are present in the sums for next state variables.

The method proposed in the last section for single input variable change flow tables can be readily extended to multiple input change flow tables by means of a technique introduced earlier by Friedman and Menon [11]. Details of this extension of the proposed state assignment are given in reference 12.

REFERENCES

1. S. H. Unger, Asynchronous Sequential Switching Circuits, Wiley-Interscience, 1969.

2. H. H. Tracey, "Internal state assignments for asynchronous sequential machines", IEEETC, EC-15, No. 4, pp. 551-560, August 1966.

3. C. N. Liu, "A state variable assignment method for asynchronous sequential switching circuits", JACM, 10, pp. 209-216, April 1963.

4. A. D. Friedman, R. L. Graham, J. D. Ullman, "Universal single transition time asynchronous state assignments", IEEETC, C-18, No. 6, pp. 541-547, June 1969.

5. R. J. Smith, II, "Generation of internal state assignments for large asynchronous sequential machines," IEEETC, C-23, No. 9, pp. 924-932, September 1974.

6. C. J. Tan, "State assignments for asynchronous sequential machines," IEEETC, C-20, pp. 382-391, April 1971.

7. D. A. Huffman, "A study of the memory requirements of sequential switching circuits," Technical Report No. 923, Research Laboratory of Electronics, MIT, March 14, 1955.

8. S. H. Unger, "A row assignment for delay-free realizations of flow tables without essential hazards," IEEETC, C-17, No. 2, pp. 146-152, February 1968.

9. D. B. Armstrong, A. D. Friedman and P. R. Menon, "Realization of asynchronous sequential circuits without inserted delay elements," IEEETC, C-17, No. 2, pp. 129-139.

10. E. B. Eichelberger, "Hazard detection in combinational and sequential switching circuits," Proc. 5th Ann. Symp. on Switching Theory and Logical Design, pp. 111-120, November 1964.

11. A. D. Friedman and P. R. Menon, "Synthesis of asynchronous sequential circuits with multiple-input changes," IEEETC, C-17, No. 6, pp. 559-566, June 1968.

12. J. G. Kuhl, "Design Techniques for Asynchronous Sequential Circuits," M. S. Thesis, Electrical Engineering, University of Iowa, Iowa City, Iowa 52242.

13. G. K. Maki and J. H. Tracey, "A state assignment procedure for asynchronous sequential circuits," IEEETC, C-20, No. 6, pp. 666-668, June 1971.

QUANTIZATION ERROR IN FLOATING-POINT ARITHMETIC*

A.B. SRIPAD
D.L. SNYDER
Washington University
St. Louis, Missouri 63130

ABSTRACT

In this paper, the behavior of the mantissa in a normalized floating-point number representation is investigated. A necessary and sufficient condition is given on a real number for its mantissa to have a reciprocal density. A statistical model to represent a floating-point quantizer with an input whose mantissa is reciprocally distributed is developed. The first- and second-order statistical properties of this model are studied.

I. INTRODUCTION

A non-zero real number x is represented in radix-2, normalized, floating-point arithmetic as $x = \pm m2^{\nu}$, where the mantissa, m, is a positive fraction satisfying $\frac{1}{2} \leq m < 1$, and where the exponent, ν, is an integer. When x is zero, both m and ν are taken to be zero. A floating-point representation has the advantage of large dynamic range compared to a fixed-point representation because ν can have many values depending on the number of bits in the register used to store it. However, there is still unavoidable quantization error when using floating-point arithmetic in practice because m must also be stored in a register having a finite number of bits. Statistical models for this quantization error have been postulated on the basis of experiments or on the basis of intuition developed from fixed-point models [1,2], and they have been used widely in digital signal processing studies.

In the statistical model, m is presumed to have a reciprocal probability density, and the error in the quantized representation of m is presumed to have a uniform probability-density [1,2]. A survey of results obtained experimentally or heuristically that motivate the reciprocal density for m is given by Knuth [3]. Hamming [4] has demonstrated the persistance of the reciprocal density for the mantissa following arithmetic operations, and, recently, Feldstein and Goodman [5] have studied the distribution of the trailing digits of the mantissa.

The main objectives of this paper are: (1), to establish a necessary and sufficient condition on a real-valued number for the mantissa in its normalized floating-point representation to have a reciprocal probability density, and (2), to determine a statistical model for the quantization-error in a floating-point quantizer if these conditions hold.

---

* This work was supported by the National Institutes of Health under Research Grant Number RR00396 from the Division of Research Resources.

II. DENSITY OF MANTISSA

Consider $\{X(i), i \in I\}$ to be a sequence of inputs to a floating-point quantizer. The normalized mantissa, M(i), and exponent, N(i), for the *ith* sample is related to X(i) according to

$$|X(i)| = M(i)\ 2^{N(i)}, \tag{1}$$

for all $i \in I$. When we study the first-order statistics of the mantissa, the index i will be dropped for notational simplication. In this paper we will consider the first-order statistics only but the details regarding the second-order statistics and the proofs of the propositions can be found in [7].

Let X be a real-valued, continuous random-variable with a probability-density function $f_X(x)$. The probability density of the mantissa $f_M(m)$ of X is of interest. As mentioned above, M is observed experimentally to have a reciprocal density. In this section, we give a necessary and sufficient condition on X for M to have such a density. This result is useful in modeling the floating-point quantizer in Section III. We then have the following two propositions.

Proposition 1:

The probability density of the mantissa, $f_M(m)$, is given by

$$f_M(m) = \begin{cases} (m \ln 2)^{-1}[1 + \sum_{n \neq 0} \phi(2\pi n) \exp(-j2\pi n \log_2 m)], & \frac{1}{2} \leq m < 1 \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

where $\phi(u)$ is the characteristic function for $\log_2|X|$.

Proposition 2:

The probability density of the mantissa $f_M(m)$ is the reciprocal density,

$$f_M(m) = \begin{cases} (m \ln 2)^{-1}, & \frac{1}{2} \leq m < 1 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

if and only if $\phi(2\pi n) = 0$ for all $n \neq 0$.

Remarks:

1) The deviation of the actual density of mantissa from the reciprocal density, if the condition of Proposition 2 is not satisfied, can be computed using Proposition 1. This measure is useful to judge the error which would be committed by use of a reciprocal-density model for the mantissa when the

necessary and sufficient condition is not satisfied.

2) There are several examples of densities for X that satisfy the condition of Proposition 2[7].

3) Second-order extensions are also made, and we show that under these conditions the mantissas form a white sequence.

4) Gaussian random variables do not satisfy the condition of Proposition 2, so the mantissa is not reciprocally distributed. Using Proposition 1 we can show that if X is Gaussian, the absolute relative error between the actual density of mantissa and (3) is upper bounded by about 0.23% independently of the standard deviation of X. Numerical evaluation of the relative error indicates that the upper bound closely approximates this error at least throughout the range of standard deviations between 0.01 and 35[7].

III. FLOATING-POINT QUANTIZER MODEL

In this section, we develop a statistical model for a rounding, floating-point quantizer when the input variable satisfies the condition of Proposition 2. We assume the grain size is $\Delta=2^{-t}$, where t is the number of bits to which the mantissa of the input is rounded.

Let $\{X_q(i)\}$ be modelled for each $i \in I$ as

$$X_q(i) = X(i)[1+E(i)], \tag{4}$$

where E(i) is the relative error between the input X(i) and output $X_q(i)$; we refer to E(i) as the quantization error. Our primary concern is the determination of the first-and second-order statistics of E. The infinite precision mantissa M of X is quantized to a number say M_q, represented by t bits, and (4) implies that

$$M_q(i) = M(i)[1+E(i)]. \tag{5}$$

In the development of statistical models for floating-point quantization-error [1,2], it is assumed that: i) M has a reciprocal probability density, ii) the error in the quantized representation of M is uniformly distributed, and, iii) the error and M are statistically independent. The assumption ii) is vacuous since a reciprocally distributed m does not satisfy the necessary and sufficient condition of the 'Quantization Theorem' [6]; further, the independence assumption iii) is unjustified. In this paper, we develop a statistical model for the quantization error, without making the above assumptions, under the condition that m has a reciprocal density.

The following Proposition is established in [7].

Proposition 3:

If the normalized mantissa has a reciprocal density, the probability density of the quantization error E for a rounding quantizer is given by

$$f_E(e) = \begin{cases} (2\Delta \ell n2)^{-1}(e+1)^{-1}, & |e| < \Delta\ (2-\Delta)^{-1} \\ (\ell n2)^{-1}[n(e)+1](e+1)^{-1}, & \Delta(2-\Delta)^{-1} \leq |e| < \Delta(1+\Delta)^{-1} \\ 0, \quad \text{otherwise}, & \end{cases} \tag{6}$$

where n(e) is the largest integer satisfying the condition

$$n(e) < (2\Delta|e|)^{-1}[\Delta-|e|(1+\Delta)].$$

Remarks:

1) If the number of bits used to represent the mantissa is so large that $\Delta=2^{-t}$ is negligible compared with unity, we deduce from (6) that the density of the quantization error is approximated by the density for E derived in [1,2]; for $t\geq 8$, the approximation is good.

2) If the conditions in Proposition 2 and its two dimensional version are satisfied, the quantization error sequence is 'white'.

IV. CONCLUSION

A model to represent the mantissa is required in order to study quantization effects in a floating-point quantizer. We have therefore investigated the distribution of the mantissa for random inputs. A necessary and sufficient condition on the input is given for the mantissa to have a reciprocal distribution; an expression to evaluate the departure of the actual density of the mantissa from the reciprocal density, if this condition is not satisfied, is also given. Further, we have shown that under suitable conditions, the mantissa is a 'white' sequence. A detailed analysis for Gaussian sequences shows that a reciprocal density model for this case is quite accurate over a wide range of standard deviations.

We have developed a model to represent a floating-point quantizer when the mantissa of the input is reciprocally distributed. As the number of bits to represent the mantissa increases, the quantizer model approaches the model that has been routinely used in earlier literature on the subject. Models to represent the quantized version of various floating-point arithmetic operations can now be developed based on our model.

REFERENCES

1. T. Kaneko and B. Liu, "On Local Roundoff Errors in Floating-Point Arithmetic," J. ACM, Vol. 20, No. 3, pp. 391-398, Jul. 1973.

2. A. Fettweiss, "On Properties of Floating-Point Roundoff Noise," IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. ASSP-22, No. 2, pp. 149-151, Apr. 1974.

3. D.E. Knuth, The Art of Computer Programming: Vol. 2, Seminumerical Algorithms, Reading, Mass.: Addison-Wesley, Ch. 4, 1969.

4. R.W. Hamming, "On the Distribution of Numbers," Bell System Tech. J., Vol. 49, No. 8, pp. 1609-1625, Oct. 1970.

5. A. Feldstein and R. Goodman, "Convergence of Estimates for the Distribution of Trailing Digits," J. ACM, Vol 23, No. 2, pp. 287-297, Apr. 1976.

6. A.B. Sripad and D.L. Snyder, "A Necessary and Sufficient Condition for Quantization Errors to be Uniform and White," To appear in the IEEE Trans. on Acoustics, Speech and Signal Processing.

7. A.B. Sripad and D.L. Snyder, "Quantization Error in Floating-Point Arithmetic," Monograph No. 323, Biomedical Computer Laboratory, Washington University School of Medicine, St. Louis, Sept. 1977.

FRACTIONAL RADIX NUMBER SYSTEMS AND THEIR APPLICABILITY TO AUTONOMOUS ONE-DIMENSIONAL DISCRETE-TIME SYSTEMS

QUENTIN L. KLEIN
Lincoln Laboratory
Bedford, MA 01730

MARTIN E. KALISKI
Department of Electrical Engineering
Northeastern University
Boston, MA 02115

This paper will discuss fractional radix number systems (FRNS's), generalizations of the standard binary number system that, in our view, provide insight into the behavior of certain one-dimensional systems.

An FRNS is a number system whose radix r is any real number obeying $1 < r < 2$ and whose digits (for the purposes of this paper) are 0 and 1. The numbers in an FRNS are of the form

$$d_j\ d_{j-1}\ \ldots\ d_1 d_0\ .\ d_{-1}\ d_{-2}\ \ldots \qquad (d_i = 0,1)$$

with the interpretation that this notation represents the real number

$$p = \sum_{k=-\infty}^{j} d_k r^{-k}$$

(when r=2 we have the conventional binary system).

It is the purpose of this paper to discuss the role that FRNS's may play in the description of the autonomous behavior of certain one-dimensional discrete-time systems. These systems are of the form

$$x_{k+1} = f(x_k) \qquad (*)$$

where f is a continuous function obeying:

(1) $f: [0,1] \rightarrow [0,1]$ ([0,1] is the unit interval)
(2) $f(0) = 0$

and

(3) There exists a point p, $0 < p < 1$, such that f is strictly increasing on [0,p] and strictly decreasing on [p,1].

This work extends previous efforts of the authors[1,2] in the theory of one-dimensional iteration.

REFERENCES

[1] Kaliski, Martin E., and Klein, Quentin L., "Toward a Theory of Dense Orbital Behavior in a Class of Autonomous One-Dimensional Nonlinear Discrete-Time Systems," Information and Control, to be published (1977).

[2] Kaliski, Martin E., and Klein, Quentin L., "Sequence Generation by Autonomous Discrete-Time Systems: Further Results," presented at 13th Annual Allerton Conference on Circuit and System Theory, 1975.

LIST SEARCHES USING MULTIPLE PROCESSORS

LEE D. CORAOR AND JOHN P. ROBINSON
Division of Information Engineering
The University of Iowa
Iowa City, Iowa 52242

The use of multiple processors to quickly search large data lists is examined. The objective is to find a match condition with given test data or to find the closest match.

The hardware organization uses an ordered list stored in independent multiple memory banks corresponding to the multiple processors. This permits independent simultaneous searches of memory by the different processors. The processors communicate with each other using a bus reporting system.

Memory is searched by the processors in parallel, accessing key index values and comparing the results simultaneously by bus reporting. A hardware generalization of interval halving results in the ordered list being partitioned not by halves but by the number of processors. All searching is done in parallel with a minimum of interprocessor communication and a minimum delay due to matching processes. Index value locations are tied to the memory address which allows for some control features to be built into the memory list organization by means of a simple calculation. Maintenance of the ordered list is needed when adding new items to the list.

Using these techniques a list with L list items and n processing elements can be searched for a match in R probes where $n^R = L$, with probe length being dependent on memory cell size. For example, with 256 processors a list of over 4 billion elements could be searched in 4 probes. For a memory cell size of 256 bits, probe length is estimated to be 100 instructions.

PROPERTIES OF A MODIFIED INCIDENCE MATRIX FOR COMPUTATION GRAPHS*

R. W. CHURCH and T. MURATA
Department of Information Engineering
University of Illinois at Chicago Circle
Chicago, Illinois 60680

Summary

Consider a digraph G in which each arc has two types of weights, y_j^+ and y_j^-. Define a modified incidence matrix of G as an n x e matrix $A = [a_{ij}]$, where n and e are the numbers of vertices and arcs in G, respectively, and

$$a_{ij} = \begin{cases} y_j^+ & \text{if j is an outgoing arc at vertex i} \\ -y_j^- & \text{if j is an incoming arc at vertex i} \\ y_j^+ - y_j^- & \text{if j is a self-loop at vertex i} \\ 0 & \text{if j is not incident with vertex i.} \end{cases}$$

The above matrix A can be used for the matrix (state) equation describing the behavior of computation graphs [1], which is similar to those for Petri nets and marked graphs [2-4]. For a computation graph, the weight y_j^+ (y_j^-) represents the number of data items added (subtracted) when an operation represented by the initial (terminal) vertex i of arc j is performed. Note that A reduces to the regular incidence matrix of a digraph (without self-loops) if $y_j^+ = y_j^- = 1$.

This paper is a preliminary investigation of the properties of the matrix A, which may play a key role in the study of the relationship between the topology and behavior of computation graphs. The incidence matrix of a connected graph G has rank n-1 and is unimodular. However, neither of these properties holds in general for the modified incidence matrix A. The following are some of the results obtained thus far.

Define the gain of arc j as $\alpha_j = y_j^+/y_j^-$. A circuit C is said to be balanced iff $\pi_{j \varepsilon C^+} a_j = \pi_{j \varepsilon C^-} \alpha_j$, where C^+ (C^-) denotes the set of arcs in C which agree (disagree) in orientation with the circuit. Otherwise, it is said to be unbalanced (a self-loop is balanced iff $y_j^+ = y_j^-$). For the case of directed circuits, the above condition for balance is equivalent to the loop gain being 1 in the terminology of [1].

Theorem 1: The determinant of the modified incidence matrix of a circuit is zero if and only if it is balanced.

Theorem 2: The rank of the modified incidence matrix of G is equal to n or n-1. A necessary and sufficient condition for the rank to be n is that

---

[*] This work was supported in part by Grant HEW PHS MB 00114.

there exists an unbalanced circuit (or unbalanced self-loop).

Theorem 3: The rank of the submatrix of the modified incidence matrix of G corresponding to a spanning tree is n-1.

It is conjectured that balanced circuits in computation graphs may play a role similar to circuits in marked graphs and thus are important for investigating properties (such as reachability) of computation graphs.

REFERENCES

[1] R. M. Karp and R. E. Miller, "Properties of a model for parallel computations: determinacy, termination, queueing", SIAM J. Appl. Math., Vol. 14, No. 6, pp. 1390-1411, November 1966.

[2] T. Murata, "State equation, controllability, and maximal matchings of Petri nets", IEEE Trans. on Automatic Control, Vol. AC-22, No. 3, June 1977.

[3] T. Murata, "Circuit theoretic analysis and synthesis of marked graphs", IEEE Trans. on Circuits and Systems, Vol. CAS-24, No. 7, July 1977.

[4] T. Murata, "Petri nets, marked graphs, and circuit-system theory -- a recent CAS application", Circuits and Systems, Vol. 11, No. 3, pp. 2-12, June 1977.

BINARY SEARCH USING UNRELIABLE COMPARISONS

R.L. Rivest, A.R. Meyer, D.J. Kleitman*
Massachusetts Institute of Technology, Cambridge, MA 02139 and
J. Spencer
State University of New York, Stony Brook, Stony Brook, N.Y.

Abstract: Searching for a value $x \in \{1,2,\ldots,n\}$ by comparing x to constants can be carried out with $\log_2(n)+k\cdot\log_2\log_2(n)+O(k)$ comparisons in the worst case even when the answers to as many as k of the comparisons may be erroneous. For k fixed independent of n, this bound is proved to be within an additive constant of optimal, even if arbitrary "Yes-No" questions about x are allowed.

I. INTRODUCTION

Let x be an unknown number which we wish to identify by asking "Yes-No" questions about x. Our objective is to minimize the number of questions required, taking into account the possibility that some of the answers received may be erroneous.

This problem comes in several versions based on the possible values of x and the nature of the "Yes-No" questions. In the discrete case x is a member of the finite set $\bar{n} =_{\text{def}} \{1,2,\ldots,n\}$; in the continuous case x is a member of the half-open real interval (0,1]. In either case, let U denote the universe of possible values of x. Independent of the choice of U, the allowable questions may in one case be arbitrary questions of the form "Is $x \in T$?", where T is a specified subset of U, or in the other case questions may be comparisons of the form "Is $x \leq a$?" where a is a specified element of U. For the continuous case we will not in general be able to identify x exactly with a given number q of questions; rather, we try to minimize the size of the region which is known to contain x. Finally, the questioning strategy may be either adaptive, where each question posed depends on the answers received to previous questions, or nonadaptive, where the questions to be asked are fixed in advance of the receipt of any answers.

We carry out a worst-case analysis of the preceding identification problems under the additional assumption that up to k of the answers we receive may be incorrect. This worst-case analysis is in contrast to the similar problem in which erroneous answers can occur randomly with given

---

* This research was sponsored by NSF Grants MCS77-19754 and MCS76-14294 and ONR Grant N000-14-76-C-0366

probability; the probabilistic problem corresponds to transmission over a noisy channel when a noiseless feedback channel is available and has been well-studied [1].

In section II we prove that, for adaptive strategies, $\lg(n)+k\lg\lg(n)+O(k)$ questions* are necessary and sufficient for the discrete case. Surprisingly, this number of questions remains sufficient even if all questions must be comparisons.

In section III we consider the continuous case when $k = 1$. A new argument is given for the lower bounds in the case of arbitrary questions, and another optimal strategy using comparisons is described.

In section IV we note that the problem of determining a small interval containing the minimum root of a set of increasing function $g_i(x)$ for $0 \le i \le k$ over the real interval $[0,1]$ by testing the signs of values of the functions reduces to a variation of the continuous case studied in section III. Our results then yield upper bounds matching those of Gal, et al. [3].

In section V we mention some open problems.

II. THE DISCRETE CASE

We first consider the arbitrary-question case. It is easy to see (as first published by Brylawski [2]) that constructing a nonadaptive q-question strategy for the discrete identification problem where up to k erroneous answers might be received reduces to the problem of constructing a k-error-correcting binary code of length q containing n code words. Conversely (as Brylawski also noted), any nonadaptive questioning strategy yields a corresponding error-correcting code. Thus, an optimal nonadaptive stategy is equivalent to an optimal code; standard coding-theory results give bounds on the parameters of the possible codes [1].

The analysis of the arbitrary-question case when an adaptive strategy may be used is more complicated. We first present a lower bound on the number of questions required. We then give a matching upper bound based on a strategy which uses only comparisons. Together these results complete the analysis of the discrete case.

Theorem 1: Let $k \ge 0$ be any fixed integer. Any (adaptive) strategy for the discrete identification problem of identifying x from $\bar{n}$ when at most k question might receive erroneous answers, must ask at least

$$\lg(n) + k \lg\lg(n) + O(1)$$

---

*We use lg to denote $\log_2$ throughout.

questions in the worst case.

Proof: An adaptive strategy can be modelled by a finite binary "decision" tree. Internal nodes correspond to questions, the left and right sons of an internal node are the nodes to be taken following a "Yes" or "No" answer respectively, and leaves are labelled with the determined value of x.

We will show that any such tree must have at least

$$\ell := (n/2)\binom{\lg(n)-1}{k}$$

leaves. Since the number of questions required in the worst case is the same as the depth of the tree, at least $\lceil \lg(\ell) \rceil$ questions are required; since k is fixed, this implies our theorem.

Lemma 1: To any leaf of depth d there corresponds a unique subset of size $\leq k$ of $\{1,\ldots,d\}$ indicating which questions on the path of the leaf were answered incorrectly.

Since the leaf determines the actual value of x, we need only compare the answers received with the correct ones for that value to retrospectively detect the lies. This proves the lemma.

Now among the n possible values of x, at most n/2 values appear at leaves of depth $\leq \lg(n)-1$, simply because in a binary tree there are at most n/2 leaves at depth $\leq \lg(n)-1$. Hence there are at least n/2 values of x which can be determined only at leaves of depth $\geq \lg(n)-1$. Since any k out of the first $\lg(n)-1$ in a series of questions which determine such a value may be answered incorrectly, and since by Lemma 1 each of the $\binom{\lg(n)-1}{k}$ subsets of incorrect answers leading to any such particular value must correspond to a distinct leaf, we conclude that there are at least ℓ distinct leaves. □

We now turn our attention to the problem of constructing efficient questioning strategies. We will show how when k is fixed to come within an additive constant of the lower bound of Theorem 1 with a strategy which only uses comparisons and not arbitrary questions. (The strategy will of necessity be adaptive; it is easy to see that any nonadaptive comparison strategy requires n-1 questions.)

Strategy D:

We first consider the case k = 1.

During the first stage of the strategy, the state of the search for $x \in \bar{n}$ can be summarized by four real numbers,

$$0 \leq L_0 \leq L_1 < R_1 \leq R_0 \leq n$$

such that

(1) $L_0 < x \le R_0$, and

(2) either $L_1 = 0$ or the question "Is $x \le L_1$?" has previously been answered "No" and

(3) either $R_1 = n$ or the question "Is $x \le R_1$?" has previously been answered "Yes".

Initially we take $L_0 = L_1 = 0$ and $R_1 = R_0 = n$. Questions asked during the first stage will be of the form "Is $x \le Z$?" for certain Z such that $L_1 \le Z \le R_1$. If the answer received is "Yes," then the values of R_1 and R_0 are updated to be Z and R_1, respectively; the values of L_0, L_1 are unchanged. Likewise, if the answer received is "No," then L_1 and L_0 are updated to be Z and L_1 respectively, and the R's remain unchanged. The reader can easily convince himself that, under the condition that at most one erroneous answer can occur, conditions (1), (2), and (3) are preserved under this updating rule.

The first stage of the strategy will terminate when $R_1 - L_1 \le 1$. Conditions (1), (2), and (3) then imply that by repeating the questions "Is $x \le L_1$?" and "Is $x \le R_1$?" it will be possible to detect an erroneous answer to an earlier occurrence of one of these questions, or else to determine that no erroneous answers occurred and hence that $x = R_1$.

In the event that an erroneous answer is detected at the end of the first stage, a second stage of the strategy commences which consists of ordinary binary search in the correct interval $(L_0, R_0]$. Since the erroneous answer has occurred before the second stage, binary search will determine the value of x.

The rule defining Z during the first stage is motivated as follows. If erroneous answers were not possible, the best strategy would be binary search in the interval $(L_1, R_1]$, namely choosing $Z = (R_1 + L_1)/2$.[†] However, if $(L_1 - L_0) >> (R_1 - L_1)$ and the "No" answer to "Is $x \le L_1$?" was erroneous, then $\lg(R_1 - L_1)$ questions would be wasted in performing binary search in the interval $(L_1, R_1]$ before the erroneous answer was detected and a proper binary search was started in the interval $(L_0, L_1]$. To minimize this waste, Z is skewed from the center of the interval $(L_1, R_1]$ by a factor which grows rapidly as $L_1 - L_0$ or $R_0 - R_1$ becomes more disproportionate to $R_1 - L_1$, so that the erroneous answer, if any, will be detected with only $\lg\lg(R_1 - L_1)$ wasted questions.

[†] We could easily restrict Z to integer values, but the analysis is more convenient when we allow arbitrary real values.

The amount by which Z is to be skewed is most easily described in terms of the quantity S defined at any point during the first stage to be the number of immediately preceding consecutive questions which received the same answer, or equivalently, the number of immediately preceding consecutive times the same pair of boundaries remain unchanged.

Define the skew factor

$$\eta(t) := \min(2^t/\lg(n),1).$$

Then

$$Z := (R_1+L_1)/2 + \eta(S)[(R_1-L_1)/2]$$

if the R's remained unchanged after the answer to the preceding question, if any, and

$$Z := (R_1+L_1)/2-\eta(S)[(R_1-L_1)/2]$$

otherwise.

We analyze strategy D for the case k = 1 in the following lemmas:

Lemma 2: During the first stage at most $\lg\lg(n)+1$ questions are ever asked between the receipt of an erroneous answer and its detection.

Proof: Assume w.l.o.g. that the question "Is $x \leq L_1$?" was answered "No" erroneously, then all subsequent questions asked during the first stage will be correctly answered "Yes." If there are $\lg\lg(n)$ such "Yes" answers, then since $\eta(\lg\lg(n)) = 1$ the strategy will then choose $Z = L_1$ and repeat the question "Is $x \leq L_1$?" thereby detecting the earlier erroneous answer. □

Lemma 3: After the first $t \geq 0$ questions of the first stage,

$$R_1-L_1 \leq n\cdot 2^{-t}\cdot(1+2/\lg(n))^t.$$

Proof: Consider two successive times during the first stage when S = 1, and suppose $r \geq 1$ questions were asked between these times. By definition of the strategy, the value of R_1-L_1 will have decreased between these two times by the factor $2^{-r}\cdot\mu(r)$ where

$$\mu(r) = (1+\eta(r))\prod_{i=1}^{r-1}(1-\eta(i)).$$

But by induction on r, it is easy to prove that

$$\mu(r) \leq (1+2/\lg(n)).$$

The lemma then follows by induction on the number of times S = 1 during the first stage. □

We remark that the proof of Lemma 3 implies that the worst cases of the first stage occur when correct "Yes" and "No" answers alternate until an erroneous answer is received.

Lemma 4: Strategy D for $k = 1$ requires at most

$$\lg(n) + \lg\lg(n) + O(1)$$

questions to identify $x \in \bar{n}$.

Proof: Suppose t_1 correct answers are received, then an erroneous answer, then t_2 further answers until the erroneous answer is detected, and finally t_3 answers until x is determined.

By Lemma 2, $t_2 \leq \lg\lg(n)+1$. Correctness of the first t_1 answers implies that $R_1 - L_1 \geq 1$ immediately after their receipt. Thus by Lemma 3, after the first t_1 answers, $1 \leq R_1 - L_1 \leq n \cdot 2^{-t_1} \cdot (1+2/\lg(n))^{t_1}$. This implies

$$t_1 \leq \lg(n) + O(1).$$

Since the first t_1 answers were correct, the t_3 answers after the lie was detected arose from the second stage binary search on the interval $(L_1, R_1]$ determined by the first t_1 answers. Hence

$$t_3 \leq \lceil \lg(n \cdot 2^{-t_1} \cdot (1+2/\lg(n))^{t_1}) \rceil = \lg(n) - t_1 + O(1).$$

The total number of questions is

$$t_1+1+t_2+t_3 \leq t_1+1+(\lg\lg(n)+1)+(\lg(n)-t_1)+O(1) = \lg(n)+\lg\lg(n)+O(1). \quad \square$$

Finally, we sketch the generalization of strategy D when $k > 1$. The general state of the search may now be summarized by numbers

$$0 \leq L_0 \leq L_1 \leq \ldots \leq L_k \leq R_k \leq \ldots \leq R_1 \leq R_0 \leq n.$$

Questions "Is $x \leq Z$?" with $L_k \leq Z \leq R_k$ are asked with Z positioned by the same rule used for the case $k = 1$. (If the answer is "No" then L_i is updated to L_{i+1} for $0 \leq i < k$ and L_k is set equal to Z. A "Yes" answer affects the R_i's symmetrically.) An erroneous answer is detected when $R_k - L_k \leq 1$. At this point the strategy is to "throw away" all information gained since (and including) the response to the earlier of the two questions "Is $x \leq L_k$?", "Is $x \leq R_k$?". By the argument used for Lemma 2 it follows that at most $\lg\lg(n)+O(1)$ questions will be thrown away at the detection of an erroneous response. The strategy now continues from the state immediately preceding the discarded answers as though for the case of $k-1$ lies. Since each erroneous answer costs $\lg\lg(n)+O(1)$ questions, and the overall strategy costs $\lg(n)+O(k)$ questions if all answers are correct, we conclude

Theorem 2: For any $k \geq 1$ and $n \geq 2$, Strategy D for identifying $x \in \bar{n}$ when up to k erroneous answers may occur, requires at most

$$\lg(n) + k\lg\lg(n) + O(k)$$

questions in the worst case.

We remark that although Strategy D is within an additive constant of optimality for any fixed k, our upper bound is significantly higher than the lower bounds we can derive for the case of k growing with n. For example, for $k = \lg(n)$, the argument of Theorem 1 implies a lower bound of only $2\lg(n)+O(1)$ whereas Theorem 2 provides an upper bound of $\lg(n)\cdot\lg\lg(n)+O(\lg(n))$.

III. THE CONTINUOUS CASE

Given a strategy for the discrete case of the problem of identifying $x'\in\bar{n}$, we can use it in the continuous case to determine an interval of length $1/n$ containing $x\in(0,1]$ by dividing the unit interval into n equal half-open subintervals. Conversely, given a strategy for the continuous case of determining an interval of size ε containing x, we can obviously use it as a strategy for the discrete identification problem of $x'\in\{1,\ldots,\lceil 1/\varepsilon\rceil\}$. Hence the results of Section II could be applied to the continuous case of determining a small interval containing x.

However, if we simply aim to determine a set of intervals of small total length, or perhaps to determine an arbitrary measurable subset of (0,1] with small measure, the proof of the lower bound of Theorem 1 does not appear to carry over. We present in Theorem 3 a general lower bound for the continuous case when k = 1.

We assume arbitrary questions of the form "Is $x\in T$?" are allowed for measurable subsets $T\subset(0,1]$.

Theorem 3: Any strategy for the continuous identification problem of determining a subset of measure $\le\varepsilon$ containing $x\in(0,1]$ when at most one question might receive an erroneous answer, must ask at least q questions in the worst case where

$$\varepsilon \ge (q+1)2^{-q} .$$

Proof: The state at any point during the questioning can be summarized by a pair of sets $A,B \subset (0,1]$ where

(1) $x\in A\cup B$, and

(2) $x\in A$ implies all preceding answers were correct, and

(3) $x\in B$ implies one preceding answer was erroneous.

Let (A_t,B_t) for $0 \le t \le q$ denote whatever state has been reached after q-t answers have been received. Thus $A_q = (0,1]$, and $B_q = \emptyset$. Let m(C) denote the measure of $C \subset (0,1]$, and define

$$w(t) := (t+1)m(A_t)+m(B_t).$$

We define an "adversary" strategy to answer the $q-t+1^{\underline{st}}$ question. Suppose this next question is "Is $x\in T$?" Let (A',B') be the next state if the answer is "Yes," so

$$(A',B') = (A_t \cap T, (B_t \cap T) \cup (A_t - T)),$$

and let $w' = tm(A') + m(B')$ be the value of $w(t-1)$ in this case. Likewise, let

$$(A'',B'') = (A_t - T, (B_t - T) \cup (A_t \cap T))$$

be the next state if the answer is "No" and let $w'' = tm(A'') + m(B'')$. The adversary answers "Yes" if $w' \geq w''$, and "No" otherwise, so that $w(t-1) = \max(w',w'')$.

Note that by trivial calculation $w'+w'' = w(t)$, and so $w(t-1) = \max(w',w'') \geq w(t)/2$. But then $\varepsilon \geq m(A_0) + m(B_0) = w(0) \geq w(q)2^{-q} = (q+1)2^{-q}$. □

We remark that the proof of Theorem 3 also yields a strategy for the continuous identification problem when $k = 1$, namely choose the next question such that $w' = w''$. By Theorem 3, such a strategy is exactly optimal in the worst case, and in fact can be chosen to be a comparison strategy.

IV. FINDING MINIMUM ROOTS

The problem of determining a small interval containing the minimum root in the unit interval of a set of increasing functions $g_i(x)$ for $0 \leq i \leq k$ is considered by Gal, et al. [3]. The small interval is to be determined by function evaluation and sign testing, that is, by asking questions of the form "Is $g_j(Z) > 0$?" for $0 \leq j \leq k$ and $Z \in (0,1]$. It is assumed that $g_i(0) < 0$ and $g_i(1) > 0$ for all i. All questions are to be answered correctly.

This minimum root identification problem arises in determining how far a point in a feasible region of a mathematical programming problem can be translated in a given direction before striking a boundary. The latter is a familiar subproblem of the common gradient-following approach to nonlinear programming (cf[3]).

The general state of the search for a minimum root can be summarized by real numbers $0 \leq L_0 \leq \ldots \leq L_k < R \leq 1$ and a permutation $\Pi:\{0,\ldots,k\} \to \{0,\ldots k\}$ such that the question "Is $g_{\Pi(i)}(L_i) > 0$?" was previously answered "No" for $0 \leq i \leq k$ and a question "Is $g_j(R) > 0$?" was previously answered "Yes" for some j. This situation is obviously analogous to the state of the continuous identification problem in a special case that "Yes" answers are never incorrect. We state informally and without proof:

Lemma 5: Any strategy for the continuous identification problem for $x \in (0,1]$ in which k erroneous "No" answers may be received (but all "Yes" answers are correct), yields an equivalent strategy for identifying the minimum root of k+1 functions.

The identification problem for $x \in (0,1]$ of Lemma 5 is a special case of the continuous identification problem of Section III, and as we observed, Strategy D of Section II can be converted to the continuous case. Thus we obtain

Theorem 4: (Gal, et al.) For any $k \geq 1$ and $n \geq 2$, there is a strategy for identifying an interval of length $\leq 1/n$ containing the minimum root in the unit interval of increasing functions $g_i(x)$ for $0 \leq i \leq k$ which requires at most

$$\lg(n) + k\lg\lg(n) + O(k)$$

function evaluations in the worst case.

We remark that the strategy we have proposed for Theorem 4 seems to be rather different from that originally given by Gal, et al.

V. CONCLUSION

Our work on the problems above is continuing along the following lines:

(1) Close the gap between the bounds of Theorems 1 and 2.

(2) Extend the optimal results of Theorem 3 for the case k = 1 to obtain an easily implementable, exactly optimal strategy for all k.

(3) Prove a converse to Lemma 5 showing that the "half-lie" problem of Lemma 5 is equivalent to minimum root identification.

(4) Extend the lower bound of Theorem 2 to the "half-lie" problem and the minimum root identification problem.

Clearly the issue of coping with erroneous answers could also be studied in the context of other familiar search problems such as sorting, finding medians, finding maxima of unimodal functions on the unit interval, etc.

VI. REFERENCES

[1] Berlekamp, E.R., "Block Coding for the Binary Symmetric Channel with Noiseless, Delayless, Feedback." in Error Correcting Codes (Wiley, 1968), pp. 61 - 85.

[2] Brylawski, T.H., "The Mathematics of Watergate," (unpublished paper).

[3] Gal, S., B. Bacherlis, and A. Ben-Tal, "On Finding the Maximum Range of Validity of a Constrained System," Technical Report #42, IBM Israel Scientific Center (Haifa, Israel, July, 1976).

[4] Katona, G.O.H., "Combinatorial Search Problems," in A Survey of Combinatorial Theory (Chapter 23), (North-Holland, 1973).

[5] Ulam, S.M., Adventures of a Mathematician (Charles Scribner's Sons, New York, 1976), page 281.

ACKNOWLEDGEMENTS

We wish to than Shimon Even, Michael J. Fischer, and Gene Lawler for helpful conversations.

We also wish to cite Ulam's memoir [5] which originally stimulated our interest in this problem.

ALGORITHM DESIGN TECHNIQUES

SARTAJ SAHNI
Computer Science Department
University of Minnesota
Minneapolis, Minnesota 55455

ABSTRACT

This paper very briefly discusses some techniques to design algorithms. A more detailed discussion may be found in [5]. The knapsack problem is used for illustrative purposes.

1. INTRODUCTION

Finding efficient algorithms to solve problems is fundamental to all computing. Let us first explore what we mean by an algorithm that solves a problem. Associated with every problem P (such as the 0/1-knapsack or traveling salesperson problems) is a function f_P from a domain D_P to a range R_P. Every point in the domain D_P specifies an instance of the problem. R_P is the set of solutions to all instances of P. The function f_P maps problem instances to solutions. f_P may be a multivalued function.

Example 1.1: In the 0/1 knapsack problem, we are provided with a knapsack of capacity M and n objects. Each object has a weight w_i and a profit p_i. We may assume w_i and p_i are greater than zero. If object i is included in the knapsack, it uses up w_i space and earns p_i profit. A solution to the knapsack problem is an n-tuple $(x_1, x_2, \ldots, x_n)$ such that $x_i = 0$ or 1, $\sum_{i=1}^{n} w_i \, x_i \le M$ and $\sum_{1}^{n} p_i \, x_i$ is maximum over all 0/1 n-tuples such that $\sum_{i=1}^{n} w_i \, x_i \le M$. Every instance of the knapsack problem may be regarded as a tuple $(M, n, w_1, p_1, w_2, p_2, \ldots, w_n, p_n)$ such that n is a positive integer and M and the w_i's and p_i's are positive. The domain, D_K, consists of all tuples of size 2n+2 where n is the second coordinate. The range, R_K, consists of all tuples with coordinate values 0 or 1. The function f_K corresponding to the knapsack problem is such that if $I = (M, n, w_1, p_1, \ldots, w_n, p_n) \in D_K$ and $S = (x_1, x_2, \ldots, x_n) \in R_K$ then $f(I) = S$ iff $\sum_{i=1}^{m} w_i x_i \le M$ and there exists no $T = (y_1, y_2, \ldots, y_n) \in R_K$ such that $\sum_{i=1}^{n} p_i y_i > \sum_{i=1}^{n} p_i x_i$ and $\sum_{i=1}^{n} w_i y_i \le M$. Note that for any given instance I, there may be many $S \in R_K$ such that $f_K(I) = S$. Hence, f_K is a multivalued function.□

An algorithm A solves a problem P if it computes f_P. Since f_P may be multivalued, it is necessary to be more precise about what we mean by A computes f_P. Two common interpretations are (i) for each instance $I \in D_P$, A computes an $S \in R_P$ such that $f_P(I) = S$ and (ii) for each instance $I \in D_P$, A computes all $S \in R_P$ such that $f_P(I) = S$. One could formulate other interpretations by limiting the number of solutions to be found.

We are now ready to formally define a problem and an algorithm for its solution. A problem P is a 4-tuple, $P = (D,R,f,I)$ where f is a function with domain D and range R. *I* is an interpretation of "computes f" (i.e., one solution, all solutions, two solutions, etc.). An algorithm A solves P iff it computes f with interpretation *I*. In order to completely specify the knapsack problem of example 1.1, we have to state the interpretation of "compute" that is desired. When no interpretation is provided, we shall assume that only one solution is desired.

In this paper, we shall be concerned mainly with a subclass of problems called optimization problems. The function f associated with an optimization problem $P = (D,R,f,I)$ is the composition of two functions f_1 and f_2, i.e., $f = f_1 o f_2$. For any instance $I \in D$ and solution $S \in R$, if $f_2(I) = S$ then S is a feasible solution to I. $f_1(S) = S$ iff $f(I) = S$, $f_1(S) = \emptyset$ otherwise. If $f_1(S) = S$ and $f_2(I) = S$ then S is an optimal solution to I. Normally, f_2 will be specified by providing a set of constraints C such that for a given instance I, $S \in R$ satisfies C iff $f_2(I) = S$. The function f_1 is specified by an objective function g such that $f_1(S) = S$ iff $g(S) = \underset{\substack{T \in R \\ f_2(I)=T}}{\text{optimum}}\{g(T)\}$ (optimum may be either max or min depending on the problem).

Example 1.2: The 0/1 knapsack problem of example 1.1 is an optimization problem. f_K may be written as $f_1 o f_2$ where for any instance $I = (M,n,w_1,p_1, \ldots, w_n,p_n)$, $S = (s_1,s_2,\ldots,s_n) \in R_K$ is feasible if and only if $\sum_{i=1}^{n} w_i s_i \le M$. This is the constraint C for this problem. This specifies f_2. $f_1(S) = S$ iff there exists no $T = (t_1,t_2,\ldots,t_n) \in R_K$ such that $f_2(I) = T$ and $\sum_1^n p_i t_i > \sum_1^n p_i s_i$. $\sum_{i=1}^{n} p_i x_i$ is the objective function. The function is optimized when it is maximum.□

In the remainder of this paper, we shall discuss some of the more common methods to obtain optimal solutions to optimization problems. These methods will be illustrated using the 0/1 knapsack problem as an example. The methods we shall discuss are: greedy method, divide and conquer, dynamic programming, backtracking and branch and bound. Other important techniques such as iteration, linear programming and integer programming are omitted.

2. THE METHODS

2.1 Greedy Method

In the greedy approach to solve a problem, an optimal solution is built up step by step. The portion of the solution that has already been built is called a partial solution. The greedy method uses an optimization measure (possibly different from the objective function) to make decisions at each step. The decision made at each step is such as to optimize the change in value of the optimization measure. Once a decision is made, it is not changed in future steps. Hence, all partial solutions must be completable to a feasible solution.

Consider how the greedy method may be used to solve the 0/1 knapsack problem. Let $I = (M,n,w_1,p_1,\ldots,w_n,p_n)$ be an instance of the 0/1 knapsack problem. A solution to the knapsack problem is completely specified by specifying which objects are to be included into the knapsack. In a greedy approach to determine these objects, we would formulate an optimization criteria based upon which, objects will be chosen for inclusion into the knapsack. At each step, a decision will be made regarding the inclusion or exclusion of exactly one object. A decision once made will not be changed. At the end of the j'th step the values of exactly j of the x_i's would have been determined. Let these j x_i's be $x_{\sigma(1)},x_{\sigma(2)},\ldots,x_{\sigma(j)}$. Let $PE(j) = \sum_{i=1}^{j} p_{\sigma(i)}x_{\sigma(i)}$ and $WU(j) = \sum_{i=1}^{j} w_{\sigma(i)}x_{\sigma(i)}$. Assume $PE(0) = WU(0) = 0$. From the requirement that all partial solutions be completable to a feasible solution, it follows that $WU(j) \leq M$, $1 \leq j \leq n$.

One possible optimization criteria that may be used in based directly on the objective function. In this criteria, at step j we attempt to include an object with highest profit. Objects are considered in nonincreasing order of p_i. Let $p_{\sigma(1)} \geq p_{\sigma(2)} \geq \ldots \geq p_{\sigma(n)}$. At step j, object $\sigma(j)$ is considered. $x_{\sigma(j)}$ is set to 1 iff $WU(j-1) + w_{\sigma(j)} \leq M$. One may verify that the resulting algorithm doesn't necessarily generate an optimal solution. In [5], other optimization measures are considered. None of these lead to an algorithm for the knapsack problem.

While the greedy method fails to result in an algorithm that solves the knapsack problem, it can be used to obtain algorithms for many other problems. [5] and [16] contain many examples of such problems. Some effort has been devoted to characterizing classes of problems that can be solved using the greedy method. One characterization is in terms of matroids [16, chapter 7].

2.2 Divide and Conquer

The basic idea in this method is to take an instance I of problem P and divide it into two or more "smaller" problem instances (the smaller problem instances may be instances of different problems). The "smaller" problem instances are solved and their solutions combined in some way to obtain an optimal solution to the original instance I. When the smaller problem instances generated are also instances of problem P then, they may be solved by recursive application of the problem division rule above.

Perhaps the simplest and most well-known application of this method is in binary search. This results in an algorithm to determine whether element x is present in the ordered set $a_1 \leq a_2 \leq \ldots \leq a_n$. Let mid = $\lfloor (n+1)/2 \rfloor$. The problem is divided into three parts: (i) is x present in $a_1,\ldots,a_{mid-1}$? (ii) is $x = a_{mid}$? and (iii) is x present in $a_{mid+1},\ldots,a_n$? The answer to (ii) is easy to obtain. The answer to one of (i) and (iii) is also easy to obtain. If $x > a_{mid-1}$ then the answer to (i) is no. Otherwise, the answer to (iii) is "no". Thus, if the answer to (ii) is "yes" the problem is solved. If the answer is "no" then we will be left with exactly one of the two subproblems (i) and (iii). This is solved by a recursive application of the problem division rule above. (Note that, usually, the comparison

$x > a_{mid-1}$ is not made. A single comparison between x and a_{mid} is made to determine whether (ii) holds and also which of (i) and (iii) must have answer "no".)

The 0/1 knapsack problem is an example of a problem in which the original problem instance is divided into two instances of a different problem [6]. The solutions from these two instances are combined to obtain a solution to the 0/1 knapsack instance. The resulting algorithm is of complexity $O(2^{n/2})$. Most other known algorithms for this problem are of complexity $O(2^n)$.

2.3 Dynamic Programming

Dynamic programming is an implicit enumeration scheme that can be used when the solution to a problem may be viewed as the result of a sequence of decisions. In §2.1 we saw how to view the solution to the knapsack problem as a sequence of decisions on the values of the x_i's. In dynamic programming an optimal sequence of decisions is arrived at by making explicit appeal to the Principal of Optimality. This principle states that an optimal sequence of decisions has the property that whatever the initial state and decision are, the remaining decisions must constitute an optimal decision sequence with regard to the state resulting from the first decision.

For any instance I of the knapsack problem, a solution may be obtained by making decisions on the values of the n variables $x_1,\ldots,x_n$. A decision on x_i involves deciding which of the values 0 or 1 is to be assigned to it. Let us assume that decisions on the x_i are made in the order $x_n,x_{n-1},\ldots,x_1$. Following a decision on x_n we may be in one of two possible states: the capacity remaining in the knapsack is M or the capacity remaining is $M - w_n$ and a profit of p_n has accrued. From the principle of optimality, it follows that the remaining decisions $x_{n-1},\ldots,x_1$ must be optimal with respect to the problem state resulting from the decision on x_n. Otherwise $x_n,\ldots,x_1$ will not be optimal.

Let $f_j(X)$ be the value of an optimal solution to the knapsack instance $I' = \{X,j,w_1,p_1,\ldots,w_j,p_j\}$ where the w_i's and p_i's are defined by I. The principal of optimality states that:

$$f_n(M) = \max\{f_{n-1}(M),\ f_{n-1}(M-w_n) + p_n\} \ldots (2.1)$$

For arbitrary $f_i(X)$, $i > 0$, equation (2.1) generalizes to:

$$f_i(X) = \max\{f_{i-1}(X),\ f_{i-1}(X-w_i) + p_i\} \ldots (2.2)$$

Equation (2.2) may be solved for $f_n(M)$ by beginning with the knowledge $f_0(X) = 0$ for all X and $f_i(X) = -\infty$, $X < 0$. $f_1,f_2,\ldots,f_n$ may be successively computed using (2.2).

Each f_i is completely specified by the pairs (P_j,W_j) where W_j is a value of X at which f_i changes. $P_j = f_i(W_j)$. If there are r changes then we need to know r pairs (P_j,W_j), $1 \leq j \leq r$. For convenience we introduce the pair $(P_0,W_0) = (0,0)$. If we assume $W_j < W_{j+1}$, $0 \leq j < r$ then from (2.2) it follows that $P_j < P_{j+1}$. Further, $f_i(X) = f_i(W_i)$ for all X such that $W_j \le X < W_{j+1}$, $0 \le j < r$. $f_i(X) = f_i(W_r)$ for all X, $X \ge W_r$. If S^{i-1} is the set of all pairs for f_{i-1} (including (0,0)) then the set S_1^i of all pairs for $g_i(X) = f_{i-1}(X - w_i) + p_i$ is obtained by adding to each pair in S^{i-1} the pair (p_i,w_i).

$$S_1^i = \{(P,W) \mid (P-p_i,W-w_i) \in S^{i-1}\} \ldots (2.2)$$

S^i may now be obtained by merging together S^{i-1} and S_1^i. This merge corresponds to taking the maximum of the two functions $f_{i-1}(X)$ and $f_{i-1}(X - w_i) + p_i$ (2.2). Thus, if one of S^{i-1} and S_1^i has a pair (P_j,W_j) and the other has a pair (P_k,W_k) and $P_j \le P_k$ while $W_j \ge W_k$ then the pair (P_j,W_j) is purged (i.e., it is discarded). This is required by (2.2). $f_i(W_j) = \max\{P_j,P_k\} = P_k$.

If (P1,W1) is the last tuple in S^n, a set of 0/1 values for the x_i's such that $\sum p_i x_i = P1$ and $\sum w_i x_i = W1$ may be determined by carrying out a search through the S^i's. We may set $x_n = 0$ if $(P1,W1) \in S^{n-1}$. If $(P1,W1) \notin S^{n-1}$ then $(P1 - p_n, W1 - w_n) \in S^{n-1}$ and we may set $x_n = 1$. This leaves us to determine how either (P1,W1) or $(P1-p_n,W1-w_n)$ was obtained in S^{n-1}. This may be done by using the argument used to determine x_n.

The algorithm which results from all of these ideas has a worst case complexity which is $O(2^n)$. This may be combined with the divide and conquer approach to obtain an $O(2^{n/2})$ algorithm.

While the technique of dynamic programming dates back to Bellman, only recently have people begun to investigate the optimality of the method for certain problems. Rosenthal ([20] and [21]) has shown the optimality of nonserial dynamic programming over a class of algorithms for discrete multivariate optimization problems in which the objective function is a sum of terms. More details on dynamic programming may be found in [2] and [18].

2.4 Backtracking

Another way to find a solution to a problem P = (D,R,f,*I*) is to first identify an easily generateable subset or superset of R in which a solution to the given instance I must lie. Next, search this subset or superset in some systematic manner for a solution. In this section we shall discuss one systematic way to search for a solution. This method is called backtracking. In further discussions, we shall assume that all solutions are expressible as n-tuples $(x_1,x_2,\ldots,x_n)$.

As mentioned earlier, for an optimization problem P, $f = f_1 o f_2$. f_2 is specified by a set of constraints. We shall divide these constraints into two categories: explicit and implicit. Explicit constraints are rules which restrict each x_i to take on values only from a given set S_i. Common explicit constraints are

$x_i \geq 0$ or S_i = {all nonnegative real numbers}
$x_i = 0$ or 1 or $S_i = \{0,1\}$
$\ell_i \leq x_i \leq u_i$ or $S_i = \{a\colon \ell_i \leq a \leq u_i\}$

The explicit constraints may or may not depend on the particular instance I of the problem being solved. All tuples that satisfy the explicit constraints define a possible solution space for I. The implicit constraints determine which of the tuples in the solution space of I are actually feasible solutions. Thus implicit constraints describe the way in which the x_i must relate to each other.

Solutions to any knapsack instance $I = (M,n,w_1,p_1,\ldots,w_n,p_n)$ are n-tuples $(x_1,\ldots,x_n)$ with $x_i = 0$ or 1, $1 \leq i \leq n$. The explicit constraint on the x_i are $x_i = 0$ or 1. The implicit constraint is that $\sum_1^n w_i x_i \leq M$.

The solution space for I consists of all the 2^n different n-tuples with coordinate values 0 or 1.

In applying backtracking to search a solution space, it is useful to organize the solution space as a tree. One possible tree organization for the knapsack problem when n=4 is shown in figure 2.1. The nodes in this tree are numbered in the order they would be generated in a depth first search.

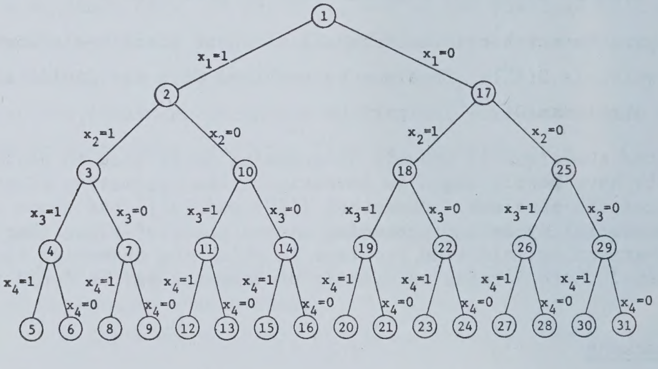


Figure 2.1 Tree organization of a solution space.

At this point it is useful to develop some terminology regarding tree organizations of solution spaces. Each node in this tree defines a problem state. All paths from the root to other nodes define the state space of the problem. Solution states are those problem states S for which the path from the root to S defines a tuple in the solution space. In the tree of Figure 2.1 only leaf nodes are solution states. Answer states are those solution states S for which the path from the root to S defines a tuple which is a member of the set of solutions (i.e., f(I) = S) to instance I. The tree organization of the solution space will be referred to as the state space tree.

Once a state space tree has been conceived of for any problem, this problem may be solved by systematically generating the problem states;

determining which of these are solution states and finally determining which solution states are answer states. There are two fundamentally different ways in which to generate the problem states. Both of these begin with the root node and generate other nodes. A node which has been generated and all of whose children have not yet been generated is called a live node. The live node whose children are currently being generated is called the E-node (node being expanded). A dead node is a generated node that is either not to be expanded further or one for which all children have been generated. In both methods of generating problem states we will have a list of live nodes. In the first of these two methods as soon as a new child of the current E-node, R, is generated, this child will become the new E-node. R will become the E-node again when all of its children nodes become dead. This corresponds to a depth first generation of the problem states. In the second state generation method, the E-node remains the E-node until it is dead. This corresponds to a breadth first generation of the problem states. In both methods, bounding functions will be used to kill live nodes without generating all their children. This will be done carefully enough so that at the conclusion of the process at least one answer node is always generated, or all answer nodes are generated if the problem requires us to find all solutions. Depth first node generation with bounding functions will be called backtracking. Breadth first (BFS) node generation methods with bounding functions will be called branch-and-bound. The branch-and-bound technique is discussed in section 2.5.

The breadth first method results in two different search strategies depending on whether a queue or stack is used to hold the live nodes. We shall reserve the terminology breadth first search to the case when a queue is used. The second alternative (i.e., use of a stack) will be call D-search.

We are now ready to present a precise formulation of the backtracking process. We will continue to treat backtracking in a general way. However, we shall assume that all answer nodes are to be found and not just one. Let $(x_1,x_2,\ldots,x_i)$ be a path from the root to a node in a state space tree. Let $T(x_1,x_2,\ldots,x_i)$ be the set of all possible values for x_{i+1} such that $(x_1,x_2,\ldots,x_{i+1})$ is also a path to a problem state. We shall assume the existence of bounding functions B_{i+1} (expressed as predicates) such that $B_{i+1}(x_1,x_2,\ldots,x_{i+1})$ is false for a path $(x_1,x_2,\ldots,x_{i+1})$ from the root node to a problem state only if the path cannot be extended to reach an answer node. Thus the candidates for position i+1 of the solution vector X(1:n) are those values X(i+1) which are generated by T and satisfy B_{i+1}. Algorithm 2.1, procedure BACKTRACK, is the general backtracking schema making use of T and B_{i+1}.

Note that T( ) will yield the set of all possible values which can be placed as the first component, X(1) of the solution vector. X(1) will take on those values for which the bounding function $B_1(X(1))$ = true. Also note how the elements are generated in a depth first manner. k is continually incremented and a solution is grown until either a solution is found or no untried value of X(k) remains. When k is decremented, the algorithm must resume the generation of possible elements for the kth position which have not yet been tried. Therefore one must develop a procedure which generates these values in some order. If only one solution is desired, a return after the print will suffice. One may easily obtain a recursive formulation for backtracking.

```
procedure BACKTRACK(n)
    //This is a program schema which describes the backtracking process.//
    //All solutions are generated in X(1:n) and printed as soon as they//
    //are determined.//
    //The function T(X(1),...,X(k-1)) generates all possible values of//
    //X(k) given that X(1),...,X(k-1) have already been chosen.//
    //The predicates $B_k$(X(1),...,X(k)) determine those//
    //elements X(k) which satisfy the implicit constraints.//
    integer k,n; local X(1:n)
    k ← 1
    while k > 0 do
       if there remains an untried X(k) such that
          X(k) ε T(X(1),...,X(k-1)) and $B_k$(X(1),...,X(k)) = true
         then if (X(1),...,X(k)) is a path to an answer node
                 then print (X(1),...,X(k)) endif
                 k ← k + 1      //consider the next set//
              endif
         else k ← k - 1      //backtrack to previous set//
       endif
      repeat
end BACKTRACK
```

Algorithm 2.1 General backtracking method

Efficiency. The efficiency of the backtracking schema we have just seen depends very much upon 4 factors: (i) the time for T to generate the next X(i); (ii) the number of X(k) satisfying the explicit constraints; (iii) the time for the bounding functions; and (iv) the number of X(k) satisfying the B_i for all i. Bounding functions are regarded as good if they substantially reduce the number of nodes that are generated. However there is usually a trade off in that bounding functions that are good also take more time to evaluate. What is desired is a reduction in the overall computing time and not just a reduction in the number of nodes generated. For many problems, the size of the state space is too large to permit the generation of all nodes. Bounding functions must be used and hopefully at least one solution will be found in a reasonable time span. Yet for many problems (e.g., n-queens) no sophisticated bounding methods are known.

If we perform a worst case analysis of backtracking programs, then the resulting formula is usually exponential. This implies that only small data sets can be fully processed. However, backtrack programs often work well on certain problem instances. Asymptotic analysis cannot really indicate whether the solution to a particular instance will be forthcoming quickly or not for centuries of computation. In order to get a better picture of the size of the solution space we turn to a restricted form of performance profiling due to Knuth [14].

The idea is based upon the strategy of sampling the solution space using some random cases. The average of those cases is taken to be the approximation to the number of nodes in the backtracking tree. This statistical method is an instance of what is referred to as a "Monte Carlo" method and somewhat surprisingly it can be shown that after only a few trials we will have obtained a reliable estimate. More details of backtracking can be found in [5] and [14].

2.5 Branch and Bound

The term branch and bound refers to all state space search methods in which all children of the E-node are generated before any other live node can become the E-node. We have already mentioned two graph search strategies, BFS and D-search, in which the exploration of a new node cannot begin until the node currently being explored is fully explored. Both of these generalize to branch and bound strategies. In branch and bound terminology, a BFS like state space search will be called FIFO (First In First Out) search as the list of live nodes is a first in first out list (or queue). A D-search like state space search will be called LIFO (Last In First Out) search as the list of live nodes is a last in first out list (or stack). As in the case of backtracking, bounding functions are used to help avoid the generation of subtrees that do not contain an answer node.

In both LIFO and FIFO branch and bound the selection rule for the next E-node is rather rigid and in a sense "blind". The selection rule for the next E-node does not give any preference to a node that has a very good chance of getting the search to an answer node quickly. We can, often, speed branch and bound by using an evaluation function $\hat{c}(\cdot)$ to evaluate live nodes. $\hat{c}(X)$ will be a measure of the "difficulty" of reaching an answer node from node X. Ideally, we would like to use a cost function $c(\cdot)$ such that for every node X, $c(X)$ is the cost of a minimum cost answer node in the subtree with root X. In case this subtree has no answer node then $c(X) = \infty$. The cost of an answer node X may be its level in the state space tree T, or the computational difficulty of reaching node X from the root T, etc. Often, it is not possible to define this ideal cost function in such a way that it may be computed easily. Instead, an estimate $\hat{c}(\ )$ for $c(\ )$ is used. This function, $\hat{c}(\ )$, is used to rank the live nodes. A live node with least $\hat{c}(\ )$ has the best chance of getting us to an answer node quickly. Hence, it makes sense to choose such a node as the next E-node. The resulting search strategy is called LC-search (Least Cost search). When bounding functions are used the search is called an LC branch and bound search.

Recently several results have been obtained concerning the efficiency of various branch and bound strategies and also on the power of bounding functions. The reader is referred to [9], [10], [11], and [15]. [19] contains a good discussion of heuristic search strategies.

2.6 Preprocessing

As the name suggests, preprocessing is concerned with performing some transformation on the input data so that the transformed problem instance is easier to solve. Many instances of this are known. Most popular amongst these is Horner's rule for evaluating a polynomial. For the case of the knapsack problem, Ingargiola and Korsh [12] have developed a scheme to reduce the problem size. The reduced problem is then solved using a standard knapsack algorithm.

2.7 Final Remark

While we have presented several different techniques to design an algorithm, it should be noted that the use of one does not preclude the use of others. In many cases, it may be necessary to combine two or more techniques to obtain an efficient algorithm.

3. CONCLUSION

By using different techniques, one can devise many different algorithms for the same problem. These may be compared both analytically and experimentally and the best determined (or a compositic algorithm made up of all or some of the different algorithms constructed). However, we will still not know whether this algorithm is best possible or even close to the best. Determining this is a very difficult problem. Non trivial lower bounds on the complexity of problems are known only for a few problems. Furthermore, these bounds are often very far from the complexity of the best known algorithms for the corresponding algorithms. For example, the best known bound for the knapsack problem is $O(n^2)$ under a restricted computation model [3]. The best algorithm has complexity $O(2^{n/2})$.

While it is difficult to determine whether an algorithm is best possible (in an asymptotic sense), it is usually not too difficult to relate a problem to other problems of similar complexity. This establishes the relative complexity of the problem. An important and well-known class of problems that are computationally related in this manner is the NP-complete class. All NP-complete problems are such that one has a polynomial time algorithm iff all the others have polynomial time algorithms. For all known NP-complete problems, the best known algorithms have a non-polynomial complexity and it is generally believed that this class of problems is not polynomially solvable. Since, it can be shown that the knapsack problem is NP-complete, we need not be overly alarmed by the fact that none of the methods of section 2 result in a polynomial time algorithm for this problem.

Sometimes, the best algorithm one can find for a problem may be too cost inefficient (as in the case of NP-complete problems). When this happens, it is worthwhile to explore other possibilities. One possibility that is gaining popularity is to obtain a fast algorithm that finds feasible solutions with value close to optimal solution values. Such algorithms are called approximation algorithms. To see the advantage of such an approach, note that for some NP-complete problems, there exist approximation algorithms of complexity $O(n/\varepsilon^2)$, $O(n^2/\varepsilon)$, etc. These algorithms guarantee feasible solutions with value within an ε fraction of the optimal for any given ε, $\varepsilon > 0$. See [4], [7], [8], [17] and [22] for a discussion of these algorithms.

A more detailed discussion of the ideas discussed in this paper appears in the forthcoming text [5].

REFERENCES

1. A. Aho, J. Ullman and J. Hopcroft, "The design and analysis of computer algorithms," Addison Wesley, 1974.
2. R. Bellman and S. Dreyfus, "Applied dynamic programming," Princeton University Press, 1962.
3. D. Dobkin and R. Lipton, "A lower bound of $\frac{1}{2}n^2$ on the knapsack problem," Yale University Technical Report, 1976.
4. M. Garey and D. Johnson, "Approximation algorithms for combinatorial problems: An annotated bibliography," in Algorithms and Complexity, Academic Press, 1976, 41-52.
5. E. Horowitz and S. Sahni, "Algorithms: Design and Analysis," Computer Science Press, 1978.

6. E. Horowitz and S. Sahni, "Computing partitions with applications to the knapsack problem," JACM, 21, 277-292 (1974).

7. E. Horowitz and S. Sahni, "Combinatorial Problems: Reducibility and Approximation," University of Minnesota, Technical Report 77-14.

8. O. Ibarra and C. Kim, "Fast approximation algorithms for the knapsack and sum of subsets problems," JACM, 22, 463-468 (1975).

9. T. Ibaraki, "Theoretical comparisons of search strategies in branch and bound algorithms, JCIS, 5(4), 315-344 (1976).

10. T. Ibaraki, "Computational efficiency of approximate branch and bound algorithms," Math. of Oper. Res., 1(3), 287-298 (1976).

11. T. Ibaraki, "On the computational efficiency of branch and bound algorithms," Jr. Oper. Res. Soc. Japan, 20(1), 16-35 (1977).

12. G. Ingargiola and J. Korsh, "A reduction algorithm for zero-one single knapsack problems, Manag. Sci., 20(4), 460-463 (1973).

13. R. Karp, "Reducibility among combinatorial problems," in Complexity of Computer Computations, Plenum Press, N.Y., 1972, 85-104.

14. D. Knuth, "Estimating the efficiency of backtrack programs," Math. of Comp., 29, 121-136 (1975).

15. W. Kohler and K. Steiglitz, "Characterization and theoretical comparison of branch and bound algorithms for permutation problems," JACM, 21, 140-156 (1974).

16. E. Lawler, "Combinatorial Optimization," Holt, Rinehart and Winston, N.Y., 1976.

17. E. Lawler, "Fast approximation algorithms for knapsack problems," Proc. 18th Ann. Symp. on Found. of Comp. Sci., Rhode Island, 1977.

18. G. Nemhauser, "Introduction to dynamic programming," John Wiley, 1966.

19. N. Nilsson, "Problem solving methods in artificial intelligence," McGraw Hill, N.Y., 1971.

20. A. Rosenthal, "The structural complexity of discrete optimization problems," Algorithms and Complexity, Academic Press, 1976, p. 490.

21. A. Rosenthal, "Nonserial dynamic programming is optimal," Proc. 9th Ann. Symp. on Theo. of Computing, Colorado, 1977, 98-105.

22. S. Sahni, "General techniques for combinatorial approximation," to appear in Op. Res.

COMBINATORIAL INFERENCE (EXTENDED ABSTRACT)

Richard DeMillo
School of Information and
Computer Science
Georgia Institute of Technology
Atlanta, Georgia

David Dobkin and Richard J. Lipton
Computer Science Department
Yale University
New Haven, Connecticut

Supported in part by ONR contract number N00014-75-C-0752
and N00014-75-0450

I. INTRODUCTION

We propose a new area of study in theoretical computer science: combinatorial inference. The basic problem is as follows:

> We have a finite set $X = x_1 \ldots, x_n$ and we wish to infer properties of elements of X on the basis of sets of "queries" regarding subsets of X.

There is an immediate and apparent distinction between combinatorial inferences and the more broadly construed kinds of logical inferences also studied in computer science. By restricting our attention to a sort of interactive dialog with a device which may deliver information concerning a finite set, we obtain problems which - *a fortiori* - concern feasible inference.

The forms of the allowable queries vary with the particular application being considered, but a great many problems in computer science, combinatorics, and optimization can be modelled in such a manner. In this paper we survery some applications of this general problem statement. We then illustrate some of the techniques available for dealing with combinatorial inference by solving some problems with particular relevance to current issues in theoretical computer science.

We begin by considering the following problems:

1. Database Security Problem [1,2,3] A database is created to contain census information concerning some subset of the population of the United States. The information is confidential and we wish to respect the privacy of individuals represented in the database. However, as the census data also contains important aggregate information about subsets of the population, we must allow serious researchers to access the aggregate information so long as no individual's is compromised.

2. Function Identification Problem [4] It is required to determine the structure of a computer program by observing selected parameters of its operation. For example, we may be given fragments of the coding of a program which computes an unknown function and we wish to determine the value of the function at a given point.

3. Group Testing Problem [5] A group of blood samples is to be processed as rapidly as possible to identify diseased persons. This is accomplished by

mixing samples to determine whether or not any members of a set of subjects is infected, so that he can be identified in future samples. However, the disease is such that sets of carriers of different strains can negate each others' effects in certain situations.

4. Physical Search [6] A guessing game is played in which the answer can be verified if found, but in which answers to queries need not be truthful.

5. Balance Problems [8,9] These are classical inference problems. One has a number of objects of some standard weight β and two objects (identical in appearance) which are defective in the sense that one weighs slightly less than β while one weighs slightly more than β by an identical amount. The defective objects are to be isolated by weighings on a three arm balance.

6. Multidimensional Search [10,11] Given a set $X = \{x_1, \ldots x_n\}$ with geometric structure we wish to determine whether or not X "contains" (i.e., determines) at point y. For instance in the case of binary search, X is a set of points on a line and y must be determined to be equal to one of the points by simple comparisons. In higher dimensions X may be a set of linear varieties, one of which may contain y.

7. Coin-Weighing Problems [12] X is chosen from $\{1, \ldots, k\}$ and the queries are of the form "what is $|S_i \cap X|$ where each $S_i \subseteq \{1, \ldots k\}$, the problem being to insure that $S_1, S_2, \ldots S_k$ determine X.

These problems may be studied by varying several parameters.

(α) Choice of Primitives Solve the indicated problem by using only queries from a suitably restricted set.

(β) Upper Bounds Determine a general strategy to insure that queries $Q_1, \ldots Q_k$ always solve the problem, where k (measured as a function of problem size and other parameters) is small.

(γ) Lower Bounds Determine minimal numbers of queries needed to infer the required properties.

(δ) Auditing Prove upper and lower bounds in the complexity of determining whether or not a given sequence of queries allow inference of a given property.

(ε) Enumeration Determine the number of "unsafe" problems for a given number of queries.

The references [1-12] consider for instance the problems (1α), (1β), (1γ), (2α), (2ε), (3β), (4β), (5ε), (6α), (6β), (6δ), (7β), (7γ), and

(7ε). Each of these is easily seen to be an instance of a combinatorial inference problem. More important, each problem instance contributes its own special flavor to the general problem. These appealing aspects of the problem domains in turn contributes to our stock of technical tools for combinatorial inference problems.

In (1), for example, x_i represents the value of some attribute of the i^{th} individual of the census population. A query might ask for the arithmetic mean of the attributes for a specified subpopulation. In general the specification of a subpopulation may be rather inhomogeneous so that it may be possible to find, for instance, the average salary of the subpopulation composed of computer scientists who are either residents of a certain county of Idaho or tenured faculty at midwestern colleges. Clearly giving the average salary of such a group is unlikely to compromise the privacy of any individual. Yet it is not at all obvious that some clever sequence of such queries couldn't yield such information. Thus if (1α), (1β) have feasible solutions, a solution to (1δ) may be required to design an "enforcer" to restrict access on questions that might compromise the salary of an individual. There are many methods for dealing with such a situation in practice. An enforcer could actually keep track of all previous queries; alternatively such a mechanism may give out information which differs slightly from that which is requested (e.g., a value may be given which is only near the true mean). These last two suggestions have been considered previously [1,2,3,13,14], and shown to be relatively ineffective since the solution to (1) allows compromise in less than the number of queries typically used by a researcher wishing to obtain legitimate information. This leads naturally to consideration of (1δ).

In interpret problem 2, let x_i be the value of a function F at point i; the problem is to infer from this value $F(y) \notin X$. For instance, we might be allowed to calculate residues of numbers modulo only certain primes and an allowable query might then consist of asking whether or not scuh a value can be found. Thus, for queries of the form "what is the value of n mod any number in $\{M_1,\ldots,M_t\}$?" a (2), (2) problem then might be how many such moduli are required to allow the value to be found, if it can be found at all.

The third problem emerges quite naturally in a number of contexts. It may be modelled as the database security problem was modelled with the proviso that it is known that the result of certain tests may be incorrect, but that these tests occur with probability less than some p, or than these outcomes occur only a bounded number of tests. Or perhaps we can determine the conditions under which the test fails and in those cases observe that one of a small set of situations can exist. Alternatively, we might be able to determine a proper set of linear combinations of samples that guarantee success of the test. This might correspond to allowing queries to be arbitrary functions of the x_i in order to find faster schemes for determining whether an individual ahs the given condition. Unfortunately, we can show that being allowed to compute any separable function of sets of data is no more valuable than being able to compute the mean to the same sets.

II. A TYPICAL PROBLEM

All of the problems of the previous section are subsumed by the following. A set X is given as the underlying database of the problem; a problem is specified by a set of primitives F and a set of restrictions R on the allowed queries. The primitives of a problem determine the legal operations to be carried out in a query while the restrictions place restrictions on the form of queries.

Examples: (i) In (1) let F consist of median $\{x_1, \ldots x_m\}$, $x_k \varepsilon X$, $m \geq t$ while R consists of the conditions that for $g \varepsilon F$, the probability that $g(X_1, \ldots, X_m)$ is $\leq q\%$ from the true median is p.

(ii) In (3), let F consist of all weighted sums of subsets of X and let R consist of failure conditions for the tests.

This is perhaps a good place to point out an interesting interpretation of problem types (β) and (δ). The problem statements "what is the least number of queries necessary to determine which of a set of blood donors is infected?" and "in how few queries can this database be compromised?" point out that upper and lower bounds can be significant for different reasons. We are often interested in worst case upper bounds and best case lower bounds.

In [1,15,3] F consisted of median operations or averages, while arbitrary selections from X were treated in [2]. In each of these studies surprisingly small upper bounds were obtained, causing us to search for restrictions to make the problem more difficult. Some apparently reasonable restrictions have been to restrict the amount of information transferred between queries (i.e., "overlap") and to restrict the number of correct versus randomly generated answers to queries. By overlap we mean that we control the number of objects that different queries can have in common. For the operations above under various restrictions the respective upper bounds are $O(\sqrt{N})$, $O(N)$, $O(N^2)$, where N is "query size". The second is optimal to within a constant factor for the indicated choices of primitives. One of the results of this paper is to present a nontrivial lower bound for median queries.

III. MAIN RESULTS

In this section we sketch some of the technical results contained in the complete paper.

We begin by considering the case in which F consists of an arbitrary selection function; i.e., for a given set of N elements from S, the F-queries select in some (unspecified) fashion one of these elements and return its value as the response to the query. More exactly, a query is defined by a set of integers $i_1, \ldots, i_n$ and a legal response is any x_j such that $j = i_m$ for some 1 m N. Note that the response does not identify "j". This type of scheme has been proposed as a practical method of ensuring database security [13]. The authors have shown [2] that if $|X| \geq N^2$, $O(N^2)$ queries are always sufficient to determine x_i for some i, even if R contains the restriction that queries overlap in at most one position. This proof depends on the existence of certain finite geometries. Now we ask whether or not this number of queries is necessary, i.e., whether a database security enforcer can always require that a user ask this number of questions in order to make the correct inference. The following result answers this question in the affirmative.

Theorem. If F consists of arbitrary selection operations on sets of $N>N_o$ indices and $|X|> N^2$ queries are required to associate i and x_i for some $i = 1,\dots|X|$, and any set of N^2-1 queries can be answered in such a way that no such inference can be made.

The key argument in this result appeals to results from transveral theory [16].

Next, we turn to problems of type (δ), i.e., we want to audit the set of queries to determine what information may be inferred on the basis of the known queries. In a given combinatorial inference problem a set of queries may be given and the problem posed to determine the strongest valid inference which can be make based on the results of the queries. Such problems arise, for instance in the blood sampling problem in which it may be desirable to determine if, after a certain set of tests are complete, the data thus obtained are sufficient to determine which of a set of donors is infected. If not, then further tests are required. Such procedures may also be used to guide the search for a good "next query".

We have been able to reduce on the inference problems of type (δ) to certain problems in matching theory as follows. Construct a bipartite graph with a vertex i for every $x_i \in X$, and a vertex r_k for every Q that returns r_k as an answer. An edge is drawn from each i to each r_k that is the result of a query containing i. In this graph an edge that belongs to all maximal matchings [17] is said to be critical. Criticality turns out to be an exact characterization of when a set of queries and results allows an inference. Determining criticality is called the offline audit problem.

Theorem. The offline audit problem for any $|X| = N$ can be solved in time at most $O(N^{3\cdot 5})$.

The offline audit problem is only one of an entire class of audit problems (δ) that can be formed by varying certain of the conditions defining the problem; e.g., there is an online (i.e., adaptive) problem formed by allowing the results of queries to be used in formulating new sets of queries.

For problems of type (α) there is quite a lot known about the behavior of inference strategies with respect to varying choices of primitives. One natural choice is to let F be composed of median queries; i.e., a query on a set of indices returns the median of the values of the indices. In work reported elsewhere, it has been shown that there is an algorithm which, when restricted to using queries of size N, can make a correct inference using only a set of queries of size $O(\sqrt{N})$. The lower bound (problem type γ) for this choice of F is not so obvious to determine. Indeed a nontrivial lower bound for median queries carries with it a great deal of practical information for the case of Problem 1, since it helps to characterize the difficulty of securing a statistical database from unwarranted extraction of information. The result which follows is established by an information-theoretic argument.

Theorem. If X is a set of rationals and F computes medians of sets of size N, then no inference problem (γ) can be solved in less than $c\log_2 N$ queries, where c is a suitable constant, independent of N.

The final category of problems considered here contains aspects which run in directions which are, in some sense, orthogonal to those considered above. The Coin Weighing Problem (7) considered in [12] is treated in several ways. First, (7δ) is solved by information-theoretic arguments. Then it is pointed out that (7β) is really composed from two problems: the one in which results of previous queries can be used to influence the choice of next query and the one in which all queries must be selected before results of queries are announced. Let us call the first strategy adaptive and the second nonadaptive. The bound (7) given in [12] holds for both adaptive and nonadaptive strategies. We have already remarked on the applicability of these concepts to the database security problems and audit problems. We turn next to the closely related Balance Problem of (5). The previously known results for this problem deal with ≤ 6 objects.

Let us define $T_A(N)$, $T_B(N)$ to be the minimal number of adaptive an non-adaptive queries needed to solve the balance problem with $|X| = N$. We have the following results:

Theorem. (1) $T_A(N) \leq 1.23 \log_2 N$
(2) $T_B(N) \leq 2 \log_3 N$
(3) $T_A(N) = \Omega(\log_2 N)$
(4) $T_B(N) = \Omega(\log_2 N)$

The lower bounds are established by rate-of-growth arguments. It is noteworthy that the optimal adaptive strategy cited in the previous theorem is the only instance of which we are aware in which an adaptive upper bound has been shown to asymptotically improve a nonadaptive upper bound.

IV. CONCLUSIONS

In this paper we demonstrate a rather broad category of problems which appear to have common formulations and which may be susceptible to the same methods of attack. The specific inferential problems which we consider give some evidence for this. In addition, the results we obtain carry some independent interest both as combinatorial results, and as results in the indicated problem domains.

REFERENCES

1. Dobkin, D., Jones, A. and Lipton, R., (to appear in ACM TODS), "Protection gainst User Inference".

2. DeMillo, R., Dobkin, D. and Lipton R., (submitted for publication), "Even Data Bases that Lie can be Compromised ".

3. Reiss, S., (1976), "A Combinatorial Model of Data Base Security", Technical Report, Yale University, Dept. of Computer Science.

4. Feldman, J.A. and Shields, P.C., (April 1972), "Total Complexity and the Inference of Best Programs", Stanford AIM-159.

5. Pippenger, N., (1976), "Group Testing", IBM Watson Research Center Technical Report.

6. Ulam, S., (1976), Adventures of a Mathematician, Scribners.

7. Katona, G.O.H., (1973), "Combinatorial Search Problems", A Survey of Combinatorial Theory, (j.N. Srivastava, Ed.), North Holland, 285-308.

8. "Shades of E 712", (November 1973), American Mathematical Monthly, Vol. 80, No. 9, 1064-1065.

9. Vilenkin, N., (1972), Combinatorial Mathematics, MIR Publishers, Moscow.

10. Dobkin, D. and Lipton R., (June 1976), "Multidimensional Searching Problems", SIAM Computing, Vol. 5, No. 2, 181-186.

11. Knuth, D.E., (April 1977), "Alorithms", Scientific American, 63-80.

12. Erdos, P. and Spencer, J., (1974), Probabilistic Methods in Combinatorics, Academic Press.

13. Conway, R. and Strip, D., (Oct. 1976), "Selective Partial Access to a Data Base", Proc. ACM 76 National Conference.

14. Denning, D. and Denning, P., Private Communication.

15. Dobkin, D., Lipton, R. and Reiss, S., "Aspects of the Data Base Security Problem", pp. 262-274, proceedings of a Conference on Theoretical Computer Science, Waterloo, Canada, 1977.

16. Mirsky, L., (1971), Transversal Theory, Academic Press.

17. Hall, M., (1967), Combinatorial Theory, Ginn and Blaisdell.

A PROBLEM IN MULTIVARIATE STATISTICS:
ALGORITHM, DATA STRUCTURE AND APPLICATIONS

Jon Louis Bentley and Michael Ian Shamos
Departments of Computer Science and Mathematics
Carnegie-Mellon University, Pittburgh, PA 15213

Abstract

We investigate problems and applications associated with computing the empirical cumulative distribution function of N points in k-dimensional space and employ a multidimensional divide-and-conquer technique that gives rise to a compact data structure for geometric and statistical search problems. We are able to show how to compute a large number of important statistical quantities much faster than was previously possible.

1. INTRODUCTION

The computational complexity of statistical procedures has just begun to be investigated [Gonzalez 77] [Shamos 77] but proves to be a rich source of new theoretical problems and algorithm design questions. In this paper we conduct an exhaustive analysis of an important computational problem in statistics involving a novel data structure that is a direct outgrowth of the divide-and-conquer algorithm used to solve the problem. We establish lower bounds on computation time, relate the problem to some searching and combinatorial questions, and present a variety of applications.

A multivariate statistical sample of N observations on each of k variables can be regarded conveniently as a set of N points in Euclidean k-space. We say that a point $X = (x_1,...,x_k)$ dominates point Y, written $X \geq Y$, iff $x_i \geq y_i$ for all i. That is, X dominates Y iff it is greater than or equal to Y in all coordinates. The dominance relation is easily seen to define a partial order on vectors. We assume for simplicity that all N points are distinct, but this will not affect the asymptotic running times of our algorithms. The rank r(Z) of a point Z (not necessarily a sample point) is the number of sample points dominated by Z. The normalized rank, r(Z)/N, which is the fraction of points dominated by Z, is better known as the empirical cumulative distribution function (ECDF) and arises in a host of statistical applications. (The use of the word "rank" in this context is suggestive but is a slight misnomer because the points are not linearly ordered. A different definition of rank is given in [Yao 74] but is unrelated to the ECDF). We now distinguish two computational problems:

1. (All points ECDF) Given N points in k-dimensional space, find the number of points dominated by each.

2. (ECDF search) Given N points in k-dimensional space, with preprocessing allowed, find r(Z) for a new arbitrary point Z (without adding Z to the data structure).

The all-points ECDF problem is the crucial step in computing the statistics associated with the Hoeffding, Kolmogorov-Smirnov (K-S), and Cramer-Von Mises tests [Hollander 73] [Hajek 67]. (These applications are discussed below.) It includes the vector maxima problem of [Kung, et al. 75] as a special case, since a maximum (in their parlance) is defined as a vector that is not dominated by any other. The reflection $Z \leftarrow -Z$ transforms a maximum into a vector whose rank is zero, so the ECDF problem can be used to find maxima.

In econometrics, it is common to represent the yield of a combination of investments as a point in multidimensional space, and one is interested in strategies that are not dominated by any others. By this view, one step in the selection of an investment portfolio is an EDCF problem.

The ECDF generalizes the notion of inversions of a permutation. Consider a two-dimensional set (x_i,y_i), such that the x_i are in increasing order. Projecting the points on the y-axis and reading them in increasing y-order induces a permutation $\pi_1,\ldots,\pi_N$ of $\{1,\ldots,N\}$. Point i is dominated by point j iff $i<j$ (that is, $x_i<x_j$) and $\pi_i<\pi_j$. (See Figure 1).

Point 5 dominates point 2 because 2 precedes 5 in the induced y-permutation
2 4 6 3 5 1

Projecting the numbered points onto the y-axis induces a permutation of 1,...,N.

Points are numbered in order of increasing x-value.

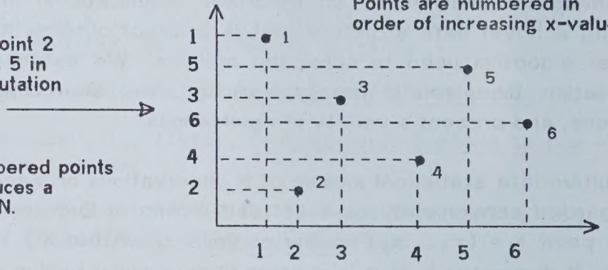


Figure 1: The connection between domination and inversions.

The number of points dominated by (x_i,y_i) is the number of inversions of π in which i participates. This shows that the ECDF is fundamentally a discrete problem and is based only on the ranks of the coordinates, not on their actual values. The generalization to higher dimensions is now clear. Let $P_1,\ldots,P_s$ be a collection of permutations of 1,...,N. Two integers i,j with $1 \le i,j, \le N$ will be said to form a s-inversion iff $i < j$ but i follows j in each permutation. For example, the pair (2,5) is a 4-inversion in the following set:

4 5 3 1 2 6   5 6 1 2 4 3   6 3 1 4 5 2   5 3 1 4 6 2

(Because 2 follows 5 in each permutation).

Determining the number of k-inversions in which each integer is involved is equivalent to a (k+1)-dimensional ECDF problem.

Finding the number of elements dominated by each member of a general partial order must take $O(N^2)$ time in the worst case but the dominance relation we have defined is a partial order of a special type -- it is induced by the lexicographic product of k linearly ordered sets. This structure will lead to a fast algorithm. This algebraic view leads to interesting combinatorial questions, such as characterizing those posets that are isomorphic to a set of points in k-space under the dominance ordering.

The ECDF is, in a very powerful sense, an excellent estimator of the underlying population CDF, which we often wish to determine. In order to be able to use this function we must be able to compute its value r(Z) at an arbitrary point Z. A typical application is the multivariate Kolmogorov-Smirnov procedure for testing the hypothesis that two samples have come from the same underlying distribution. The test statistic K is equal to the maximum absolute difference between the ECDFs of the two samples. To compute K it suffices to evaluate the ECDF of sample A at each of the points in sample B, and vice-versa. This entails a search for each point of B to determine how many points in A it dominates. Below we discuss a number of search algorithms that illustrate various time-space tradeoffs and concentrate on one that runs in $O(\log^2 N)$ time (in two dimensions) and requires only O(N log N) space and preprocessing time.

2. ECDF Algorithms

It is easy to solve the all-points ECDF problem in $O(kN^2)$ time by comparing each of the N points against every other point to determine how many it dominates. While in a general partial order of N-vectors this would also be a trivial lower bound on the time needed to compute the number of vectors dominated by each, we have seen that the dominance ordering is of special form. In the present case we will use this structure to improve on the naive algorithm.

We employ a multidimensional divide-and-conquer scheme similar to the one described in [Bentley 76a] and [Bentley 76b], which at each level of recursion reduces the dimension of the problem by one and the number of points by a factor of two.

Theorem 1: The all-points ECDF problem can be solved in $O(N \log^{k-1} N)$ time in the worst case.

Algorithm:

1. Let P be a hyperplane, normal to one of the coordinate axes, that divides the collection of points into two subsets A and B, each containing N/2

points. Such a plane can be found in O(N) time by choosing P to pass through a point that has median first coordinate.

2. Recursively solve the all-points problem on subsets A and B. That is, for each point in A we find the number of points in A that it dominates, and similarly for B. If T(N,k) is the time required to solve the entire problem, then the solution of these two subproblems can be accomplished in time 2T(N/2,k).

3. Without loss of generality, let A be the set of points whose first coordinate does not exceed that of any point of B. Note that the ECDF values obtained for set A in the recursive subproblem solution are the correct final values, since P was constructed so that no point of A can dominate any point of B. It remains only to update the B values to reflect the number of points in A that are dominated.

4. We now observe that each point of B already dominates each point of A in at least one coordinate, namely, the coordinate whose axis is normal to the dividing plane P. This coordinate can thus be removed from further consideration in forming the corrected solution for B. The coordinate can be eliminated without changing any dominance relations by merely projecting all of the points onto P, which is a subspace of one lower dimension. This projection can be accomplished in O(N) time if pointers are used instead of copying lists of coordinates. (Otherwise, O(kN) time would be required.) The projected subproblem can be solved in time T(N,k-1). Note that the "subproblem" is of a somewhat special form, as we are only interested in learning for each point of B the number of points in A that it dominates.

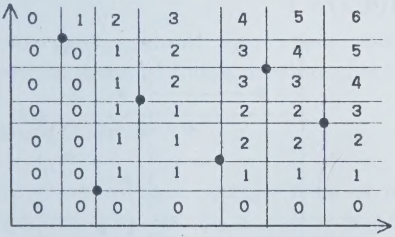
5. Combining the subproblem solutions obtained in steps 2 and 4 can be accomplished in O(N) time, so the recurrence relation for T is
$$T(N,k) = 2T(N/2,k) + T(N,k-1) + O(N) \quad .$$
By sorting the points in advance on each coordinate we may make use of the fact that T(N,2) = O(N log N) [Shamos 77] to obtain
$$T(N,k) = O(N \log^{k-1} N) + O(kN \log N) \quad .$$

In unusual circumstances, the number of dimensions may greatly exceed the number of sample points, N, in which case the above recursion is inefficient because its effort is concentrated on reducing the number of points in the subproblems. If $k > N^4$, the method of [Yao 74], which explicitly constructs the matrix of the partial order, can be used to compute the ECDF in $O(kN^2/\log N)$ time. If only the vector maxima are desired, a slight modification of the above algorithm achieves the $O(N \log^{k-2} N)$ performance attained (for k>2) in [Kung 75]. It has been shown [Bentley 77a] [Bentley 77b] that this modified maxima algorithm runs in O(N) expected time for a very wide class of input distributions.

3. ECDF Searching

Once the all-points problem has been solved, we are in a position to arrange the solution into a data structure that will make it easy to determine the number of points dominated by a new point X. The most simple algorithm requires no preprocessing at all and operates by comparing X to each of the N k-dimensional sample points. This obvious approach requires O(kN) initialization time, O(kN) query time, and O(kN) storage. It is somewhat surprising that the query time can be significantly reduced with no asymptotic increase in the storage used. The method of k-d trees [Bentley 75] achieves $O(kN^{1-1/k})$ search time after O(kN log N) preprocessing, but still needs only O(kN) storage [Lee 77].

It is possible to perform ECDF searching extremely rapidly if sufficient storage and preprocessing time are available. The method is based on the fact that there exist k-dimensional rectangular parallelepipeds within which the number of dominated points remains constant. We may easily see why this is true. Consider some point Z (not in the original set) for which r(Z) = 5 and imagine moving Z parallel to some coordinate axis. (Refer to Figure 2.)



0	1	2	3	4	5	6
0	0	1	2	3	4	5
0	0	1	2	3	3	4
0	0	1	1	2	2	3
0	0	1	1	2	2	2
0	0	1	1	1	1	1
0	0	0	0	0	0	0

Horizontal and vertical lines are drawn through each sample point. Within each of the resulting rectangles the number of dominated points (and hence the ECDF) is constant.

Figure 2: The ECDF is constant within rectangular regions.

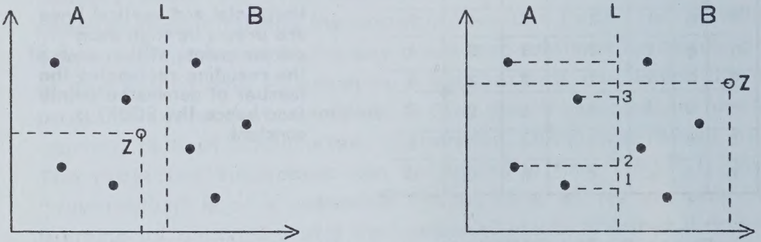
The value of r(Z) cannot change until Z passes the projection of some point of the original set on that axis. This is true of each coordinate. If we were to construct hyperplanes normal to the coordinate axes at each sample point (a total of Nk hyperplanes), these would divide space into $(N+1)^k$ rectangular regions within each of which the function r is constant. Such a structure can readily be queried in O(k log N) time by a binary search along each coordinate, so we have

Theorem 2: ECDF searching can be performed in O(k log N) time, with $O(N^k + kN \log N)$ storage and preprocessing time.

The storage used by this procedure is prohibitive. We propose a data structure and search algorithm that nearly achieves O(log N) search time, but which uses less than quadratic storage. The data structure is "isomorphic" to the all-points ECDF algorithm in the sense that it is a tree structure having a branch corresponding to each recursive call in the algorithm. The storage required by the search procedure and the time used by the all-points algorithm are described by exactly the same recurrence relation. Furthermore, the data structure for searching can be built conveniently during the solution of the all-points problem at no asymptotic increase in running time.

Let us first treat the two-dimensional case (refer to Figure 3). The dividing line L is the two-dimensional instance of the hyperplane P described in the all-points algorithm, and is used to define the first test. Let A be the set of N/2 points to the left of line L and let B be the set of points to the right. Given a new point Z we want to determine the number of points (in both A and B) that it dominates. In a single comparison against L we can determine whether Z lies in A or in B. If Z lies in A (the diagram on the left in Figure 3) it cannot possibly dominate any point of B, so we may confine our attention to a subproblem of half the size of the original. The recurrence describing this situation is just

$$T(N) = T(N/2) + 1 \ .$$



If Z lies to the left of L we need only find its rank in A.

If Z lies to the right we need its rank in B and its y-rank in A.

Figure 3: The two cases of ECDF searching in the plane.

If we learn from the first comparison that Z lies in B then the problem is only slightly more complicated (the right diagram in Figure 3). We must find the number of points in B that are dominated by Z, which can be done in time T(N/2). We then add to that the number of points in A dominated by Z. Since, however, the x-coordinate of Z is known to be greater than that of point of A, this number is merely the number of points of A that lie below Z. If we project the points of A onto L and sort them in advance (as part of the preprocessing) we will be able to locate Z in this ordering in O(log N) time by binary search. Thus the recurrence that results when Z is in B is

$$T(N) = T(N/2) + O(\log N) \ .$$

It is immediate that $T(N) = O(\log^2 N)$, even if the second case arises after each comparison.

The generalization to k dimensions is completely straightforward. The line L is replaced by a hyperplane and the sorted list by a (k-1)-dimensional ECDF search structure. The search time is given by the recurrence

$$T(N,k) = T(N/2,k) + T(N,k-1), \qquad T(N,1) = O(\log N),$$

of which the solution is $T(N,k) = O(\log^k N)$. The storage requirement of this algorithm is easy to analyze in view of its recursive structure. In two dimensions we need to store two data structures on N/2 points and one linear list of length N/2. Thus, $S(N,2) = 2S(N/2,2) + O(N) = O(N \log N)$. In k dimensions, we have

$$S(N,k) = 2S(N/2,k) + S(N/2,k-1) = O(N \log^{k-1} N) \ .$$

The preprocessing time is described by precisely the same relation, giving

Theorem 3: ECDF searching can be accomplished in $O(\log^k N)$ time, using $O(N \log^{k-1} N)$ storage and $O(N \log^{k-1} N)$ time for preprocessing.

4. Applications

We now will present some new applications of the ECDF algorithms and elaborate on some of those presented in the introduction.

4.1 Range Queries

An inconvenient but common type of geometric search problem is the range query [Knuth 73]. Given a set of N points in the plane, with preprocessing allowed, how many lie in the rectangle defined by $a \le x \le b$ and $c \le y \le d$? Within the framework of the ECDF problem, range searching becomes elementary. In two dimensions, (see Figure 4) we may find the number of points lying in rectangle ABCD by computing

$$\mathrm{ECDF}(A) - [\ \mathrm{ECDF}(B) + \mathrm{ECDF}(C)\ ] + \mathrm{ECDF}(D) \ .$$

This is a simple application of the combinatorial principle of inclusion-exclusion. We are thus able to respond to a range query by evaluating the distribution function at four points.

In k dimensions we need to evaluate the ECDF at 2^k points, but this still only requires $O(\log^k N)$ time and $O(N \log^{k-1} N)$ storage (or, $O(2^k \log N)$ time and $O(N^k)$ storage.) The range query example provides a link between the empirical distribution function and the empirical density function. The fraction of a set contained in a plane region F is a consistent estimator of the probability content of that region (the probability density integrated over F) [Loftsgaarden 65].

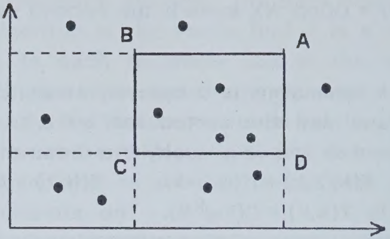


Figure 4: Range searching as an ECDF problem.

4.2 Kolmogorov-Smirnov Statistic

Because of its intimate relation to the ECDF, we point out an anomaly between the K-S one- and two-sample tests. The K-S one-sample statistic is the maximum deviation between the ECDF of a finite point set and a given hypothetical CDF (which we assume can be evaluated at a single point in constant time). A linear-time one-sample K-S algorithm in one dimension has been given by [Gonzalez 77]; it makes use of the fact that any CDF must be a monotonic function. While the situation in higher dimensions is unclear, the ECDF algorithm of this paper can be used to compute the K-S statistic in $O(N \log^{k-1} N)$ time. The K-S two-sample statistic is the maximum deviation between the ECDFs of two given finite point sets.

Theorem 4: The Kolmogorov two-sample statistic must take O(N log N) time to compute, in the worst case.

Proof: The K-S two-sample statistic is zero iff the point sets are identical. Set equality is shown to require O(N log N) comparisons in [Reingold 72].

Acknowledgements

We wish to thank Jerome Friedman of the Stanford Linear Accelerator and Lawrence Rafsky of the Chase Manhattan Bank for concentrated and helpful discussions. The algorithms described here were implemented by Elizabeth Rentmeesters at Carnegie-Mellon University and at SLAC.

Summary

We see that the empirical cumulative distribution function, a ubiquitous quantity in statistical analysis, can be computed quickly at the given sample points and can be evaluated quickly at other points. The data structure for ECDF searching was arrived at directly from the ECDF algorithm itself. The problems we have considered impinge on many others in different applications areas, all of which may be solved by the techniques developed here.

References

[Bentley 75] Bentley, J.L. Multidimensional Binary Search Trees Used for Associative Searching. CACM 18,9(1975), pp. 509-517.

[Bentley 76a] Bentley, J.L. Divide and Conquer Algorithms for Closest Point Problems in Multidimensional Space. PhD. Thesis. Univ. of North Carolina at Chapel Hill (1976).

[Bentley 76b] Bentley, J.L. and Shamos, M.I. Divide-and-Conquer in Multidimensional Space. Proc. Eighth Annual ACM Symposium on Theory of Computing. Hershey, PA (1976), pp. 220-230.

[Bentley 77a] Bentley, J.L. and Shamos, M.I. Divide and Conquer for Linear Expected Time. Information Processing Letters, to appear.

[Bentley 77b] Bentley, J.L, Kung, H.T., Schkolnick, M., and Thompson, C. On the Average Number of Maxima in a Set of Vectors, this Proceedings.

[Gonzalez 77] Gonzalez, T., Sahni, S., and Franta, W.R. An Efficient Algorithm for the Kolmogorov and Lilliefors Tests. ACM TOMS March, 1977.

[Hajek 67] Hajek, J. and Sidak, Z. Theory of Rank Tests. Academic Press (1967). 297 pp.

[Hollander 73] Hollander, M. and Wolfe, D. Nonparametric Statistical Methods. Wiley(1973). 503pp.

[Knuth 73] Knuth, D.E. The Art of Computer Programming. Volume III: Sorting and Searching. Addison-Wesley (1973).

[Kung 75] Kung, H.T., Luccio, F., and Preparata, F.P. On Finding the Maxima of a Set of Vectors. JACM 22(1975), 469-476.

[Lee 77] Lee, D.T. and Wong, C.K. Worst-Case Analysis for Region and Partial Region Searches in Multidimensional Binary Search Trees and Quad Trees. IBM T.J. Watson Research Center Preprint (1977).

[Loftsgaarden 65] Loftsgaarden, D.O. and Quesenberry, C.P. A Nonparametric Estimate of a Multivariate Density Function. Ann. Math. Stat. 36(1965), pp. 1049-1051.

[Reingold 72] Reingold, E.M. On the Optimality of Some Set Algorithms. JACM 19,4(1972), pp. 649-659.

[Shamos 77] Shamos, M.I. Geometry and Statistics: Problems at the Interface. In Algorithms and Complexity, J.F. Traub, ed. Academic Press (1977).

[Yao 74] Yao, A. C.-C. and Yao, F.F. On Computing the Rank Function of a Set of Vectors. University of Illinois Computer Science Department Report UIUCDCS-R-75-699 (1975).

COUNTER MACHINES AND NUMBER-THEORETIC PROBLEMS

OSCAR H. IBARRA
Department of Computer Science, University of Minnesota
Minneapolis, Minnesota 55455

ABSTRACT

We look at decision questions concerning two-way multicounter machines whose input and/or counters are reversal-bounded and show their relationships to some number-theoretic problems.

INTRODUCTION

A desirable property of the class of finite automata that is not shared by other well known classes of devices is the existence of decision procedures for all interesting questions concerning finite automata [20]. For example, the following problems which we shall refer to as D-problems are decidable: emptiness, infiniteness, disjointness, containment, universe, and equivalence problems. (These are the problems of deciding for arbitrary machines M_1 and M_2 in the class whether $T(M_1) = \emptyset$, $T(M_1)$ is infinite, $T(M_1) \cap T(M_2) = \emptyset$, $T(M_1) \subseteq T(M_2)$, $T(M_1)$ is the set of all finite-length strings, and $T(M_1) = T(M_2)$, respectively, where $T(M_i)$ denotes the language accepted by M_i.) The proof of the decidability of these questions makes use of the fact that a finite automaton has only a finite number of internal states or configurations. When a read-write unbounded memory is attached to the automaton, almost all D-problems become unsolvable. If the unbounded memory is in the form of a semi-infinite tape which is operated sequentially we get a Turing machine (TM). For TM's the D-problems are undecidable [3,14]. If the tape is restricted to operate on a last-in-first-out basis, the device reduces to a pushdown machine, and the emptiness and infiniteness problems become solvable [2]. All other decision questions are unsolvable [2]. For example, the containment and disjointness problems for deterministic pushdown machines are undecidable [6], and the results carry over to the restricted cases when the pushdown store is reversal-bounded (i.e., finite-turn) or when the pushdown store has a single-letter alphabet (i.e., it is operated as a counter). The status of the equivalence problem is still open although it has been shown decidable for the restricted cases just mentioned [22,23]. For nondeterministic counter machines, the universe problem is unsolvable even if the counter is restricted to make at most one reversal [1]. Perhaps the simplest known subclass of deterministic pushdown machines is the simple deterministic pushdown machines studied in [16]. Even for this subclass, the containment problem is undecidable [5]. It has, however, a decidable equivalence problem [16].

In this paper we exhibit a large class of machines for which the D-problems are decidable. A class that immediately comes to mind is the family of deterministic counter machines. However, as we have already pointed out, the disjointness and containment problems are unsolvable for these machines. Moreover, their unsolvability remain even if the machines are restricted to operate in real time. This led us to put a restriction on the counter. Initially, we looked at deterministic counter machines that can only make a bounded (i.e., finite) number of reversals on the counter. We found that all the D-problems are decidable. We then generalized the model by allowing it to have two-way input and several counters [1,4,8], but restrict it to operate in such a way that in any accepting computation, the device makes a bounded number of reversals on the counters as well as on the input. Again, we are able to show that the D-problems are decidable for these generalized machines. If the bounded-reversal restriction on either the input or the counters is removed, then the D-problems become unsolvable. For example, the class of deterministic machines with unrestricted two-way input and two reversal-bounded counters has unsolvable D-problems. (The proof uses the undecidability of Hilbert's tenth problem [17].) However, if the input alphabet is restricted to one-letter, then every deterministic two-way multicounter machine whose counters are reversal-bounded can effectively be reduced to an equivalent (ordinary) finite automaton. Thus for such machines the D-problems are solvable, and this result can be used to show the solvability of some number-theoretic problems.

We shall use the following notation: DC(k,m,n)[NC(k,m,n)] will denote the class of deterministic [nondeterministic] two-way finite-state acceptors augmented by k counters. Each machine in the class operates in such a way that in every accepting computation the count in each counter alternately increases and decreases by at most m times and the input head reverses direction by at most n times. When we have no known finite bound for the counter reversal or input reversal, we use $m = \infty$ or $n = \infty$. Thus, e.g., DC(1,∞,0) is equivalent to the class of deterministic one-way one-counter machines.

Complete proofs for the results presented here can be found in [9,10, 11,15]. However, for most results, we give an indication of how the proof runs.

THE CLASSES DC(k,m,n) AND NC(k,m,n)

Every recursively enumerable set can be accepted by a machine in

DC(2,∞,0). This is a well known result due to Minsky [18]. The proof in [18] also shows the unsolvability of the emptiness problem for the class DC(1,∞,∞). However, if the input and counters are reversal-bounded, we have the following result.

Theorem 1. The emptiness, infiniteness and disjointness problems for the class NC(k,m,n) are decidable.

Proof. The proof is in [15]. It involves showing the following: (1) The family of languages accepted by machines in $\bigcup_{k,m,n\geq 1}$NC(k,m,n) is effectively closed under union and intersection; (2) Every machine in NC(k,m,n) can effectively be converted into an equivalent machine in NC(k',1,0) for some k'; (3) The Parikh mapping [19] of the language accepted by any machine in NC(k',1,0) is an effectively computable semilinear set. (The theorem follows from (1), (2) and (3) since the emptiness and infiniteness problems for semilinear sets are decidable [7]).

DC(k,m,n) is effectively closed under complementation [15]. Then from Theorem 1 we have

Theorem 2. The D-problems for the class DC(k,m,n) are decidable.

The universe problem is undecidable for the class of nondeterministic one-way one-counter machines whose counter makes at most one reversal [1]. Hence, Theorem 2 does not hold for NC(1,1,0).

In the remainder of this section we shall look at the effect of removing the bounded-reversal restriction on either the input or the counters.

Theorem 3. The D-problems are undecidable for the following classes of machines: (a) DC(2,∞,0), (b) DC(1,∞,∞), (c) DC(1,∞,1).

Proof. The unsolvability of the D-problems for classes (a) and (b) follows from the result of Minsky [18] while that of (c) is shown in [15].

For completeness, we also state the following result which was implicitly proved in [12].

Corollary 1. The disjointness problem for the class DC(1,∞,0) is undecidable.

Next, we show that dropping the restriction on input reversals makes the D-problems unsolvable.

Theorem 4. The D-problems are undecidable for the following classes of machines: (a) $\bigcup_{m\geq 1}$DC(2,m,∞), (b) $\bigcup_{k\geq 1}$DC(k,1,∞).

The proof of Theorem 4 uses the undecidability of Hilbert's tenth problem [17]. Hilbert's tenth problem [13] is the problem of deciding for a given polynomial $P(x_1,\ldots,x_n)$ with integer coefficients (i.e., a Diophantine polynomial), whether it has an integral root, i.e., integers $\alpha_1,\ldots,\alpha_n$ such that $P(\alpha_1,\ldots,\alpha_n) = 0$. We shall restrict $x_1,\ldots,x_n$ to assume only nonnegative integers since $P(x_1,\ldots,x_n)$ has a root in the integers if and only if one of the 2^n polynomials obtained by replacing some variables by their negative values has a root in the nonnegative integers.

Theorem 4 follows from the undecidability of Hilbert's tenth problem [17] and the following theorem.

Theorem 5. The emptiness problem for machines in $\bigcup_{m\geq 1} DC(2,m,\infty)$ that accept bounded languages (i.e., languages that are subsets of $a_1^*\ldots a_r^*$ for some $r \geq 1$ and distinct symbols $a_1,\ldots,a_r$) is decidable if and only if Hilbert's tenth problem is decidable.

Proof. Let $P(x_1,\ldots,x_n)$ be a Diophantine polynomial. Let $P(x_1,\ldots,x_n) = t_1(x_1,\ldots,x_n) + \ldots + t_u(x_1,\ldots,x_n)$ where each $t_j(x_1,\ldots,x_n)$ is a term. For each $1 \leq j \leq u$, let $t_j(x_1,\ldots,x_n) = s_j x_1^{i_{j1}}\ldots x_n^{i_{jn}}$, where s_j is $+$ or $-$, $i_{j1},\ldots, i_{jn} \geq 0$. Let $a_1,\ldots,a_n,b_1,\ldots,b_u$ be distinct symbols. Define the language $L = \{a_1^{\alpha_1}\ldots a_n^{\alpha_n} b_1^{\beta_1}\ldots b_u^{\beta_u} \mid \alpha_1,\ldots,\alpha_n \geq 0,\ \beta_j = \alpha_1^{i_{j1}}\ldots\alpha_n^{i_{jn}},\ s_1\beta_1 + \ldots + s_u\beta_u = 0\}$. We can effectively construct a machine M in $DC(2,m,\infty)$ accepting L, where $m = \sum_{j=1}^{u} (i_{j1} + \ldots + i_{jn})$. (The construction of M is described in [15].) Clearly $L \neq \emptyset$ if and only if $P(x_1,\ldots,x_n)$ has a solution in the nonnegative integers. Thus, the solvability of the emptiness problem implies the solvability of Hilbert's tenth problem. The proof of the converse is in [11].

We are unable to prove Theorem 4 for the class $\bigcup_{m\geq 1} DC(1,m,\infty)$. There does not seem to be a way to reduce Hilbert's tenth problem to the emptiness problem for this class. If, on the other hand, the emptiness problem is solvable, the proof would probably not involve semilinear sets since there are languages accepted by machines in $DC(1,1,\infty)$ (e.g., $L = \{a^m b^{km} \mid k,m \geq 1\}$) whose Parikh mappings are not semilinear. However, we can prove the following result which relates the emptiness problem to the existence of a nonnegative integral solution to a system of Diophantine equations of a special form.

Theorem 6. Let $m \geq 1$. The emptiness problem for machines in $DC(1,2m-1,\infty)$ that accept bounded languages is decidable if and only if it is decidable to determine for a given system S of m equations in m + n variables $(n \geq 1)$ of the form

$$y_i = \frac{\sum_{j=1}^{n} a_{ij}x_j + b_i}{\sum_{j=1}^{n} c_{ij}x_j + d_i} \qquad i = 1,2,\ldots,m \qquad a_{ij}, b_i, c_{ij}, d_i \text{ nonnegative integers}$$

whether there exist nonnegative integers $y_1,\ldots,y_m,x_1,\ldots,x_n$ satisfying S.

Proof. Let S be a system of m equations in m + n variables of the form shown above. Let $e_1,\ldots,e_n$ be distinct symbols. Define the language $L = \{e_1^{x_1}\ldots e_n^{x_n} \mid$ there are nonnegative integers $x_1,\ldots,x_n,y_1,\ldots,y_m$ satisfying S$\}$. Clearly, we can construct a machine M in DC(1,2m - 1,∞) accepting L, and $L \neq \emptyset$ if and only if S has a nonnegative integral solution. Thus, if the emptiness problem is decidable then so is the problem of deciding if a system S has a nonnegative integral solution. The converse is proved in [11].

If we allow the coefficients a_{ij}, b_i, c_{ij}, d_i to be arbitrary integers, Theorem 6 takes the following form:

Theorem 6'. The emptiness problem for machines in $\bigcup_{m\geq 1} DC(1,m,\infty)$ that accept bounded languages is decidable if and only if it is decidable to determine for a given system S of r equations in r + n variables $(n \geq 1)$ of the form

$$y_i = \frac{\sum_{j=1}^{n} a_{ij}x_j + b_i}{\sum_{j=1}^{n} c_{ij}x_j + d_i} \qquad i = 1,2,\ldots,r \qquad a_{ij}, b_i, c_{ij}, d_i \text{ arbitrary integers}$$

whether there exist nonnegative integers $y_1,\ldots,y_r,x_1,\ldots,x_n$ satisfying S.

In [21] it is shown that determining whether a polynomial in several variables with degree ≤ 2 has a nonnegative integral solution is decidable. Theorem 6 then gives the following corollary.

Corollary 2. The emptiness problem for machines in DC(1,1,∞) that accept bounded languages is decidable.

The equivalence problem for the class DC(1,∞,0) is solvable [23]. However, the disjointness and containment problems are unsolvable. This follows from Corollary 1 and the fact that the class of languages defined by DC(1,∞,0) is closed under complementation.

THE CLASS DC(k,m,∞) OVER A ONE-LETTER INPUT ALPHABET

By Theorem 4, the emptiness problem is unsolvable for deterministic two-way 2-counter machines whose counters are reversal-bounded, even if the inputs are restricted to come from $a_1^* \ldots a_r^*$ ($r \geq 1$ each a_i a distinct symbol.) Also, as we remarked earlier, Minsky's result [18] shows that the emptiness problem for two-way one-counter machines over a 1-letter input alphabet is undecidable. In this section, we consider machines in DC(k,m,∞) whose input alphabet is restricted to one letter. Call this class DC1(k,m,∞). (Thus, a machine in DC1(k,m,∞) is a deterministic two-way k-counter machine over a one-letter input alphabet whose counters make at most m reversals.)

Machines in DC1(k,m,∞) can be used to solve some number-theoretic problems. For example, it is easy to construct a machine M_1 in DC1(2,1,∞) that accepts an input of the form 1^x (with end markers) if and only if the following system of equations holds for some positive integers x, y_1 and y_2:

$$\frac{1}{3}x - y_1 = 0$$
$$x\left\lceil \frac{y_1}{2} \right\rceil + 5\left\lfloor \frac{y_1}{2} \right\rfloor - 3y_2 = 0$$

As another example, consider a polynomial $p(x) = a_1 x^k + \ldots + a_k x + a_{k+1}$ with rational coefficients. One can construct a machine M_2 in DC1(2,m,∞) for some m such that M_2 when given input 1^x, calculates and stores in one of the 2 counters the value $\lfloor \mathrm{abs}(p(x)) \rfloor$. Moreover, M accepts 1^x if and only if $p(x)$ is a nonnegative integer.

For the class DC1(k,m,∞), we can prove the following theorem [9].

Theorem 7. Let M be in DC1(k,m,∞). We can effectively construct a two-way finite automaton M' accepting the same language as that of M. Hence the D-problems for the class DC1(k,m,∞) are decidable.

Proof. The proof in [9] consists of showing that the input head position and the values of the counters immediately after a change in the "mode" of operation of any counter can be represented by polynomials with rational coefficients whose degrees cannot exceed some fixed positive integer d. The rational coefficients come from a fixed finite set which depends only on M. The two-way finite automaton M' equivalent to M successively constructs the polynomial representations of the input head position and counter values at instances immediately after counter-mode changes.

Theorem 7 can be used to show the solvability of some number-theoretic problems. We have seen 2 examples already. As a third example, consider a system of nonlinear Diophantine equations of the form S: $A\bar{y} = \bar{\sigma}(x)$, where A is a $m \times n$ matrix of rational constants and $\bar{y} = (y_1, \ldots, y_n)$ and $\bar{\sigma}(x) = (\sigma_1(x),$

$\ldots,\sigma_m(x))$ are column vectors, each $\sigma_i(x) = r_i(x)$ or $\lfloor r_i(x) \rfloor$, $r_i(x)$ being a rational function of x with rational coefficients. Let $Q(S) = \{(y_1,\ldots,y_n,x) \mid y_1,\ldots,y_n,x$ are nonnegative integers satisfying $S\}$. Then we have

Theorem 8. It is decidable to determine for a given system S: $A\bar{y} = \bar{\sigma}(x)$ (of the form shown above) whether Q(S) is empty, finite or infinite.

Proof. Given a system S, we can construct a deterministic two-way multicounter machine M whose counters are reversal-bounded accepting the set $L(S) = \{1^x \mid$ there exist nonnegative integers $y_1,\ldots,y_n,x$ such that $A\bar{y} = \bar{\sigma}(x)\}$. The construction of M is given in [9]. Clearly $L(S) = \emptyset$ if and only if $Q(S) = \emptyset$. Thus the emptiness problem for Q(S) is solvable. If L(S) is infinite, which is decidable, then Q(S) is infinite. If L(S) is finite, then the members of L(S) can be effectively found since L(S) is a regular set. Let $x_1,x_2,\ldots,x_r$ be the members of L(S). Now consider the systems of equations, S_i: $A\bar{y} = \bar{b}_i, \bar{b}_i = (\sigma_1(x_i),\ldots,\sigma_m(x_i))$, $i = 1,2,\ldots,r$. The set $Q(S_i)$ is an effectively computable semilinear set [24]. Hence, it is decidable if $Q(S_i)$ is finite or infinite. Since Q(S) is infinite if and only if $Q(S_i)$ is infinite for some $1 \leq i \leq r$, it follows that the infiniteness problem for Q(S) is decidable.

Remark. [10] contains another proof of the solvability of the emptiness problem for Q(S). It is also shown in [10] that the emptiness problem is NP-complete when restricted to systems S satisfying a certain condition on encoding. Essentially, the restriction is for the maximum degree of the polynomials defining the $\sigma_i(x)$'s to be some polynomial of the length of the encoding of S.

REFERENCES

1. Baker, B. and Book, R., Reversal-bounded multipushdown machines, Journal of Computer and System Sciences, 8(1974), 315 - 332.
2. Bar-Hillel, Y., Perlis, M. and Shamir, E., On formal properties of simple phrase structure grammars, Zeitschrift für Phonetik Sprachwissenshaft und Kommunikationsforschung, 14(1961), 143 - 172.
3. Davis, M., "Computability and Unsolvability," McGraw-Hill, New York 1958.
4. Fischer, P., Meyer, A. and Rosenberg, A., Counter machines and counter languages, Mathematical Systems Theory, 2(1968), 265 - 283.
5. Friedman, E., The inclusion problem for simple languages, Theoretical Computer Science, 1(1976), 297 - 316.
6. Ginsburg, S. and Greibach, S., Deterministic context-free languages,

Information and Control, 9(1966), 620 - 648.

7. Ginsburg, S. and Spanier, E., Bounded Algol-like languages, Transactions of the American Mathematical Society, 113(1964), 333 - 368.
8. Greibach, S., Remarks on the complexity of nondeterministic counter languages, Theoretical Computer Science, 1(1976), 269 - 288.
9. Gurari, E. and Ibarra, O., A decidable class of counter machines with applications to some number-theoretic problems, CSci Tech. Report No. 77-10, University of Minnesota.
10. Gurari, E. and Ibarra, O., A NP-complete number-theoretic problem, CSci Tech. Report No. 77-12, University of Minnesota.
11. Gurari, E. and Ibarra, O., Two-way counter machines and Diophantine equations, in preparation.
12. Hartmanis, J. and Hopcroft, J., What makes some language theory problems undecidable, Journal of Computer and System Sciences, 4(1970), 368 - 376.
13. Hilbert, D., Mathematische Probleme. Vortrag, gehalten auf dem internationalen Mathematiker-Kongrass zu Paris 1900, Nachr. Akad. Wiss. Göttingen Math.-Phys. (1900) 253 - 297; Translation: Bull. Am. Math. Soc., 8(1901 - 1902), 437 - 479.
14. Hopcroft, J. and Ullman, J., "Formal Languages and their relation to automata," Addison-Wesley, Reading, Mass., 1969.
15. Ibarra, O., Reversal-bounded multicounter machines and their decision problems, to appear in the Journal of the Association for Computing Machinery.
16. Korenjak, A. and Hopcroft, J., Simple deterministic languages, IEEE Conference Record of 7th Annual Symposium on Switching and Automata Theory (1966), 36 - 46.
17. Matijas'evič, J., Enumerable sets are Diophantine, Soviet Math. Dokl., 2(1970), 354 - 357.
18. Minsky, M., Recursive unsolvability of Post's problem of Tag and other topics in the theory of Turing machines, Annals of Math., 74(1961), 437 - 455.
19. Parikh, R., On context-free Languages, Journal of the Association for Computing Machinery, 13(1966), 570 - 581.
20. Rabin, M. and Scott, D., Finite automata and their decision problems, IBM Journal of Research and Development, 3(1959), 114 - 125.
21. Siegel, C., Zur theorie der quadratischen formen Nachr. Akad. Wiss. Gottingen Math.-Phys. Kl. II(1972), 21 - 46.
22. Valiant, L., The equivalence problem for deterministic finite-turn pushdown automata, Information and Control, 25(1974), 123 - 133.

23. Valiant L. and Patterson, M., Deterministic one-counter automata, Journal of Computer and System Sciences, 10(1975), 340 - 350.

24. Van Leeuwen, J., A partial solution to the reachability problem for vector addition systems, Proceedings of Sixth Annual ACM Symposium on Theory of Computing, Seattle, Washington, 1974, 303 - 309.

ON THE ESTIMATION OF DISCRETE PROBABILITY DENSITIES FROM NOISY MEASUREMENTS

LUC P. DEVROYE
School of Computer Science
McGill University
Montreal, Canada H3C 3G1

GARY L. WISE
Department of Electrical Engineering
University of Texas at Austin
Austin, Texas 78712

ABSTRACT

The estimation of a discrete probability density from independent observations is considered. For a wide class of noises, a method is given for estimating a probability density when the measurements are corrupted by independent additive noise. This method is shown to be consistent, and several bounds on the error are given.

I. INTRODUCTION

The need for considering discrete data is often encountered in data communications, digital signal processing, and other areas. In this paper we consider discrete valued random variables, and we are concerned with estimating the discrete probability density function. Measurements are taken, and from these measurements a density function is obtained. However, we assume that the measurements are imperfect. We derive the estimators, establish the appropriate forms of convergence, and supply an abundance of bounds on the errors.

Assume that we can observe $X_1, X_2, \ldots, X_n$, a sequence of independent identically distributed random variables with the unknown discrete probability density f. An obvious way of estimating f(x) is to use the empirical density based on the n observations. However, the estimation problem is complicated if we can only observe $X_1 + Z_1, X_2 + Z_2, \ldots, X_n + Z_n$, where $X_1, Z_1, X_2, Z_2, \ldots, X_n, Z_n$ are independent random variables and the Z_i's, commonly referred to as noise, have a common known discrete probability density function g. For a wide class of densities g, a method is given to recover f which is shown to be strongly uniformly consistent, that is,

$$\lim_{n\to\infty} \sup_x \left| f_n(x) - f(x) \right| = 0$$

with probability one (wp1), where f_n is the estimate of f with just n observations.

II. PROPERTIES OF THE EMPIRICAL DENSITY

Let $X_1, X_2, \ldots, X_n$ be independent identically distributed random variables with a discrete probability density function f. Assume without loss of generality that f is supported on $\mathbb{Z}$, the set of integers. The empirical density f_n is defined by

$$f_n(x) = \frac{1}{n} \sum_{i=1}^{n} I_{\{X_i = x\}}, \quad x \in \mathbb{Z},$$

where I is the indicator function. Thus, f_n is also a discrete probability density.

We will briefly review some properties of f_n, starting with the pointwise consistency. By the strong law of large numbers [1, p.239] we know that $f_n(x) \to f(x)$ wp1 for all x. In fact, by Hoeffding's inequality [2], for all $\varepsilon > 0$,

$$P\left\{ \left| f_n(x) - f(x) \right| \geq \varepsilon \right\} \leq 2 \exp(-2n\varepsilon^2),$$

from which the strong consistency follows by the Borel-Cantelli lemma [1, p.228]. This bound is independent of f and x. Since countable unions of null events are null, we immediately have the strong uniform consistency

$$\lim_{n\to\infty} \sup_{x} |f_n(x) - f(x)| = 0 \quad \text{wp1}.$$

Now we consider some uniform error bounds. If F_n and F are the distribution functions corresponding respectively to f_n and f, then

$$P\left\{\sup_{x} |f_n(x) - f(x)| \geq \varepsilon\right\} \leq 2P\left\{\sup_{x} |F_n(x) - F(x)| \geq \frac{\varepsilon}{2}\right\}$$

$$\leq 2C_1 \exp\left[-2n\left(\frac{\varepsilon}{2}\right)^2\right], \tag{1}$$

by an inequality of Dvoretzky, Kiefer, and Wolfowitz [3], where C_1 is a universal constant. In the Appendix it is shown that $C_1 < 610.4$. Recently, Singh [4] (see also, [5]) has shown that

$$P\left\{\sup_{x} |F_n(x) - F(x)| \geq \beta\right\} \leq 4\, e^2\, n\, \beta \exp\left(-2n\beta^2\right)$$

for $n\beta^2 \geq 1$. This implies that

$$P\left\{\sup_{x} |f_n(x) - f(x)| \geq \varepsilon\right\} \leq 4\, e^2\, n\, \varepsilon \exp\left(-n\, \frac{\varepsilon^2}{2}\right) \tag{2}$$

for $n\varepsilon^2 \geq 4$. Both bounds (1) and (2) are valid for all discrete densities f and, by the Borel-Cantelli lemma, each implies that

$$\lim_{n\to\infty} \sup_{x} |f_n(x) - f(x)| = 0 \quad \text{wp1}.$$

Now consider the following two lemmas.

Lemma 1: Let $f_n(x)$ be the empirical estimate of the discrete density f(x). Then

$$P\left\{\sum_{x=-\infty}^{\infty} |f_n(x) - f(x)| \geq \varepsilon\right\} \leq K_1 \exp(-K_2 n)$$

where $K_1, K_2 > 0$ depend upon ε and f only.

Proof: Pick $N \geq 1$ such that $\sum_{|x|>N} f(x) < \frac{\varepsilon}{6}$. Then

$$P\left\{\sum_{x=-\infty}^{\infty} |f_n(x) - f(x)| \geq \varepsilon\right\}$$

$$\leq P\left\{\sum_{|x|\leq N} |f_n(x) - f(x)| \geq \frac{\varepsilon}{2}\right\} + P\left\{\sum_{|x|>N} |f_n(x) - f(x)| \geq \frac{\varepsilon}{2}\right\}$$

$$\leq \sum_{|x|\leq N} P\left\{|f_n(x) - f(x)| \geq \frac{\varepsilon}{4N+2}\right\} + P\left\{\sum_{|x|>N} f_n(x) - \sum_{|x|>N} f(x) \geq \frac{\varepsilon}{6}\right\}$$

$$\leq (4N+2)\ \exp\left[-2n\left(\frac{\varepsilon}{4N+2}\right)^2\right] + \exp\left[-2n\left(\frac{\varepsilon}{6}\right)^2\right]$$

$$\leq (4N+3)\ \exp\left[-2n\left(\frac{\varepsilon}{4N+2}\right)^2\right]$$

by Hoeffding's inequality [2]. Q.E.D.

Lemma 2: Let $(\Omega,\mathscr{B},P)$ be a probability space. Let f_1, f_2, ... be densities on the integers for each fixed ω, and random variables on $(\Omega,\mathscr{B},P)$ for each fixed x. We will write $f_n(x,\omega)$ to make the dependency on ω explicit. Let f be a density on the integers. If $f_n(x,\omega) \overset{n}{\to} f(x)$ wpl for all x, then

$$\sum_{x=-\infty}^{\infty} |f_n(x,\omega) - f(x)| \overset{n}{\to} 0 \quad \text{wpl.} \tag{3}$$

Proof:

$$P\left\{\omega:\ \overline{\lim_{n\to\infty}} \sum_{x=-\infty}^{\infty} |f_n(x,\omega) - f(x)| \neq 0\right\} \leq$$

$$P\left\{\omega:\ \sum_{x=-\infty}^{\infty} \overline{\lim_{n\to\infty}} |f_n(x,\omega) - f(x)| \neq 0\right\} =$$

$$P\left\{\bigcup_{x=-\infty}^{\infty}\left(\overline{\lim_{n\to\infty}} |f_n(x,\omega) - f(x)| \neq 0\right)\right\} \leq$$

$$\sum_{x=-\infty}^{\infty} P\left\{\omega:\ \overline{\lim_{n\to\infty}} |f_n(x,\omega) - f(x)| \neq 0\right\} .$$

Therefore, if $f_n(x,\omega) \to f(x)$ wpl, then

$$\sum_{x=-\infty}^{\infty} |f_n(x,\omega) - f(x)| \to 0 \quad \text{wpl.}$$ Q.E.D.

III. ESTIMATION IN THE PRESENCE OF ADDITIVE NOISE

Because of background noise, faulty equipment, or other practical problems, it may not be possible to observe X_1, X_2, ..., X_n; but instead, we can observe Y_1, Y_2, ..., Y_n, where

$$Y_i = X_i + Z_i , \quad 1 \leq i \leq n ,$$

and X_1, Z_1, X_2, Z_2, ..., X_n, Z_n are independent. The Z_i have a common known density g on the integers and the X_i have an unknown density f on the integers which we would like to estimate. The discrete probability density h of the Y_i is given by

$$h(x) = \sum_{y=-\infty}^{\infty} f(x-y)\ g(y) .$$

Assume that we can write

$$f(x) = \sum_{y=-\infty}^{\infty} \xi_y\ h(x-y) = \sum_{y=-\infty}^{\infty} \xi_{x-y}\ h(y)$$

for some sequence $\{\xi_i\}$ of real numbers. Resubstitution gives

$$f(x) = \sum_{y=-\infty}^{\infty} \xi_y \sum_{u=-\infty}^{\infty} g(u)\, f(x-y-u) \tag{4}$$

which should hold for all x and all f.

Lemma 3: Eq. (4) is valid for all x and all f if and only if

$$\sum_{y=-\infty}^{\infty} \xi_y\, g(k-y) = \begin{cases} 1 & \text{for } k = 0 \\ 0 & \text{for } k \neq 0 \end{cases}, \tag{5}$$

for all integers k.

Proof: Clearly, if (5) holds, then (4) is valid for all x and all f. Conversely, let f(0) = 1, and note that

$$\sum_{y,u:y+u=0} \xi_y\, g(u) = 1 \quad .$$

Next, let $f(0) = \frac{1}{2} = f(k)$, $k \neq 0$. Then (4) reads, for x = 0,

$$\frac{1}{2} = \frac{1}{2} \sum_{y,u:y+u=0} \xi_y\, g(u) + \frac{1}{2} \sum_{y,u:y+u=-k} \xi_y\, g(u)$$

$$= \frac{1}{2} + \frac{1}{2} \sum_{y,u:y+u=-k} \xi_y\, g(u) \quad ,$$

from which (5) follows by the arbitrariness of k. Q.E.D.

Deferring for the moment the question of how to determine the ξ_y from g so that (5) is satisfied, we return to the construction of an estimate of f assuming the knowledge of the ξ_y. Let h_n be the empirical density for $Y_1, Y_2, \ldots, Y_n$,

$$h_n(x) = \frac{1}{n} \sum_{i=1}^{n} I_{\{Y_i=x\}} \quad ,$$

which suggests the following estimate of f:

$$f_n(x) = \sum_{y=-\infty}^{\infty} \xi_y\, h_n(x-y) = \sum_{y=-\infty}^{\infty} \xi_{x-y}\, h_n(y)$$

$$= \frac{1}{n} \sum_{i=1}^{n} \sum_{y=-\infty}^{\infty} \xi_{x-y}\, I_{\{Y_i=y\}} = \frac{1}{n} \sum_{i=1}^{n} \xi_{x-Y_i} \quad . \tag{6}$$

Notice that for g with g(0) = 1 (and thus $\xi_0 = 1$, $\xi_i = 0$, $i \neq 0$), we get back the original empirical estimate of f because

$$\xi_{x-Y_i} = I_{\{Y_i=x\}} \quad .$$

The first question that arises is the question of the closeness of f_n to f. Notice that f_n is not a probability density in general. Of course, $\tilde{f}_n$ defined by

$$\tilde{f}_n(x) = \begin{cases} 1 & , \quad f_n(x) \geq 1 \\ f_n(x) & , \quad 0 < f_n(x) < 1 \\ 0 & , \quad f_n(x) \leq 0 \end{cases}$$

is a strictly better density estimate than f_n. However, we will not further discuss this trivial modification of our estimate. Clearly, f_n satisfies

$$\sup_x |f_n(x) - f(x)| \leq \left(\sum_{y=-\infty}^{\infty} |\xi_y| \right) \sup_x |h_n(x) - h(x)| \quad , \tag{7}$$

$$\sup_x |f_n(x) - f(x)| \leq \left(\sup_y |\xi_y| \right) \sum_{x=-\infty}^{\infty} |h_n(x) - h(x)| \quad , \tag{8}$$

and

$$\sum_{x=-\infty}^{\infty} |f_n(x) - f(x)| \leq \left(\sum_{y=-\infty}^{\infty} |\xi_y| \right) \sum_{x=-\infty}^{\infty} |h_n(x) - h(x)| \quad .$$

Let

$$C = \sum_{y=-\infty}^{\infty} |\xi_y|$$

and

$$D = \sup_y |\xi_y| \quad .$$

Applying some of the results of the previous section, we have the following theorem.

Theorem 1: Let $f_n(x)$ be given by (6) and assume that $\{\xi_y\}$ satisfies (5). Then if C is finite, we have

$$P\left\{ \sup_x |f_n(x) - f(x)| \geq \varepsilon \right\} \leq \begin{cases} 2\, C_1 \exp\left[- \dfrac{n\, \varepsilon^2}{2\, C^2} \right], & \text{all } n \geq 1 \\[2ex] \dfrac{4e^2 n\varepsilon}{C} \exp\left[- \dfrac{n\, \varepsilon^2}{2\, C^2} \right], & \text{all } n \geq \dfrac{4C^2}{\varepsilon^2} \end{cases} \tag{9}$$

and

$$P\left\{ \sum_{x=-\infty}^{\infty} |f_n(x) - f(x)| \geq \varepsilon \right\} \leq K_1 \exp(-K_2 n) \tag{10}$$

where $K_1, K_2 > 0$ are constants depending upon ε, h, and C. Also, if D is finite, we have

$$P\left\{\sup_x |f_n(x) - f(x)| \geq \varepsilon\right\} \leq K_3 \exp(-K_4 n) \qquad (11)$$

where K_3, $K_4 > 0$ are constants depending upon ε, h, and D.

The bounds in (9) are distribution-free but require that $C < \infty$. Eq. (11), which is not distribution-free requires only that $D < \infty$. The strong result (10) assumes finiteness of C and is not uniform over all densities.

Using the Borel-Cantelli lemma, we obtain the following result.

Corollary 1: Let $f_n(x)$ be given by (6) and assume that $\{\xi_y\}$ satisfies (5). If $D = \sup_y |\xi_y| < \infty$, then

$$\sum_{x=-\infty}^{\infty} |f_n(x) - f(x)| \xrightarrow{n} 0 \quad \text{wp1.}$$

In the remainder of this paper we briefly discuss practical solutions to (5) and give some examples of sequences $\{\xi_y\}$ for some common densities g.

Practical Considerations: A solution to (5) can be obtained recursively if g is a single tailed density, that is, if there exists a K such that $g(x)=0$ for all $x > K$ or $g(x) = 0$ for all $x < K$. For example, assume that $g(K) > 0$ and $g(x) = 0$ for all $x < K$. Let $\xi_y = 0$ for $y < -K$ and $\xi_{-K} = 1/g(K)$. It is easy to see that the $k = 0$ equation of (5) holds and that the $k = 1$ equation results in

$$\xi_{-K+1}\, g(K) + \xi_{-K}\, g(K+1) = 0 \quad ,$$

from which we find ξ_{-K+1}. Solving the $k = 2$ equation of (5) gives us ξ_{-K+2} and so on. Clearly, this is probably not the only solution to (5). Consider the following simple example.

Example 1: Let $g(0) = g(1) = 1/2$ and $g(x) = 0$, $x \neq 0, 1$. Then (5) results in

$$\xi_{-1} + \xi_0 = 2$$

$$\cdots = \xi_{-2} + \xi_{-1} = \xi_0 + \xi_1 = \xi_1 + \xi_2 = \cdots = 0 .$$

If $\xi_0 = \alpha$ and $\xi_{-1} = 2 - \alpha$, then all solutions of (5) can be written as

$$\xi_y = \alpha(-1)^y \quad , \ y \geq 0$$

$$\xi_{-y} = (2 - \alpha)(-1)^{y+1} \quad , \ y \geq 1$$

where α is any real number. For this case we note that D is finite while C is infinite.

Example 2: In Example 1 we have $\sup_y |\xi_y| < \infty$, but this is not always the case. If $g(-1) = g(1) = 1/4$ and $g(0) = 1/2$, then it is straightforward to show that for any solution of (5) we have $\sup_y |\xi_y| = \infty$.

Now we will give solutions for some well known densities g.

Example 3: Poisson noise with parameter $\lambda > 0$: Let

$$g(x) = \frac{\lambda^x}{x!} e^{-\lambda} \quad , \quad x \geq 0 \quad ,$$

and it can be verified that a solution to (5) is given by

$$\xi_y = \begin{cases} \frac{(-\lambda)^y}{y!} e^{\lambda} & , \quad y \geq 0 \\ 0 & , \quad y < 0 \end{cases} .$$

It is easily seen that

$$\sum_{y=-\infty}^{\infty} |\xi_y| = e^{2\lambda} < \infty \quad .$$

In [6] the estimation of a continuous probability density function from measurements corrupted by Poisson noise is considered.

Example 4: Geometric noise with parameter $\lambda > 1$: If

$$g(x) = (\lambda-1)/\lambda^{x+1} \quad , \quad x \geq 0 \quad ,$$

then a solution to (5) is given by

$$\xi_y = \begin{cases} \lambda/(\lambda-1) & , \quad y = 0 \\ -1/(\lambda-1) & , \quad y = 1 \\ 0 & , \quad y \neq 0, 1 \end{cases} .$$

Example 5: Binomial noise with parameters N and $p \neq 0, 1$: If

$$g(x) = \binom{N}{x} p^x (1-p)^{N-x} \quad , \quad 0 \leq x \leq N \quad ,$$

then a solution to (5) is given by

$$\xi_y = \begin{cases} (-1)^y \binom{N+y-1}{y} (1-p)^{-N} \left(\frac{p}{1-p}\right)^y & , \quad y \geq 0 \\ 0 & , \quad y < 0 \end{cases} .$$

We notice that

$$\sum_{y=-\infty}^{\infty} |\xi_y| < \infty$$

if and only if $p < 1/2$, and that $|\xi_y| \xrightarrow{y} \infty$ if $p > 1/2$ or if $N > 2$ and $p = 1/2$. Another solution to (5) is given by

$$\xi_{-N-y} = \begin{cases} (-1)^y \binom{N+y-1}{y} p^{-N} \left(\frac{1-p}{p}\right)^y & , \quad y \geq 0 \\ 0 & , \quad y < 0 \end{cases}$$

and it is easy to see that for each $p > 1/2$,

$$\sum_{y=-\infty}^{\infty} |\xi_y| < \infty \quad .$$

APPENDIX

In this appendix we prove the following Lemma. The bound that we present results directly from the work of Dvoretzky, Kiefer, and Wolfowitz.

Lemma: If F is any distribution function on $\mathbb{R}$ and F_n is the empirical distribution function with $X_1, X_2, \ldots, X_n$, a sequence of independent random variables with distribution function F, then

$$P\left\{\sup_x |F_n(x) - F(x)| \geq \varepsilon\right\} \leq C_1 \exp\left(-2n\varepsilon^2\right)$$

where

$$C_1 = 2\left[1 + \frac{32}{3\sqrt{6\pi}} + \frac{4}{3\sqrt{3}} + \frac{\exp(40/9)}{8\sqrt{2e}}\right] .$$

Proof: We will use the notation of Dvoretzky, Kiefer, and Wolfowitz [3, pp.646-648]. In [3] they establish that

$$1 - G_n(r) \leq 2\left[1 - H_n(r)\right]$$

where r takes values in $(0, \sqrt{n}\ )$. Expression (2.9),

$$1 - H_n(r) = \left(1 - \frac{r}{\sqrt{n}}\right)^n + r\sqrt{n} \sum_{j=[r\sqrt{n}]+1}^{n-1} Q_n(j,r) \quad ,$$

can be upper bounded as follows. Notice that

$$\left(1 - \frac{r}{\sqrt{n}}\right)^n < \exp\left(-2\ r^2\right)$$

(see (2.11)).

First, consider those j for which $|j - \frac{n}{2}| \leq \frac{n}{4}$. We will show that

$$Q_n(j,0) < c_2\ n^{-3/2}$$

for $c_2 = \dfrac{16}{3\sqrt{6\pi}}$. By an approximation of Feller [7] for n! we have

$$\begin{aligned} Q_n(j,0) &= \binom{n}{j} \frac{j^j}{n^n} (n-j)^{n-j-1} \\ &\leq \frac{1}{\sqrt{2\pi}} \sqrt{\frac{n}{j(n-j)^3}} \exp\left[\frac{1}{12n} - \frac{1}{12j+1} - \frac{1}{12(n-j)+1}\right] . \end{aligned}$$

Notice that $j(n-j)^3 \geq 27 \left(\frac{n}{4}\right)^4$. Also,

$$\exp\left[\frac{1}{12n} - \frac{1}{12j+1} - \frac{1}{12(n-j)+1}\right] < \exp\left[\frac{-1}{12j+1}\right] < 1 .$$

Thus,

$$Q_n(j,0) < \frac{16}{3\sqrt{6\pi}} n^{-3/2} .$$

Next, we know that

$$\frac{1}{n} + \int_0^\infty \exp\left(-8 r^2 t^2\right) dt < 1 + \frac{\sqrt{2\pi}}{4}$$

and (2.15) holds:

$$r\sqrt{n} \sum{}' Q_n(j,r) < 2 c_2 \left(1 + \frac{\sqrt{2\pi}}{4}\right) \exp\left(-2 r^2\right)$$

$$= \left(\frac{32}{3\sqrt{6\pi}} + \frac{4}{3\sqrt{3}}\right) \exp\left(-2 r^2\right) .$$

Now consider those j for which $|j - \frac{n}{2}| > \frac{n}{4}$. It follows from the equation at the top of p.647 that

$$Q_n(j,r) \leq Q_n(j,1) \exp\left[-f(n,r,j) + f(n,1,j)\right]$$

for $r \geq 1$, where $f(n,r,j)$ is the negative of the exponent in (2.12). From (2.9) it follows that $j \geq r\sqrt{n}$. Thus, it is easily seen that $f(n,1,j) \leq 40/9 = c$. Therefore, we have that

$$Q_n(j,r) \leq c\, Q_n(j,1) \exp\left[-f(n,r,j)\right]$$

for $r \geq 1$. Since $f(n,r,j) \geq 2 r^2 + \frac{r^2}{64}$, we have that

$$Q_n(j,r) \leq c\, Q_n(j,1) \exp\left(-2 r^2\right) \exp\left(- \frac{r^2}{64}\right) .$$

Using the fact that $r \exp\left(- \frac{r^2}{64}\right)$ is maximized at $r = \sqrt{2}/16$, we obtain

$$Q_n(j,r) \leq \frac{c}{8r\sqrt{2e}} Q_n(j,1) \exp\left(-2 r^2\right) .$$

Thus,

$$r\sqrt{n} \sum{}'' Q_n(j,r) \leq \frac{c}{8\sqrt{2e}} \sqrt{n} \exp\left(-2 r^2\right) \sum{}'' Q_n(j,1)$$

$$< \frac{c}{8\sqrt{2e}} \exp\left(-2 r^2\right)$$

from (2.16).

Collecting bounds, we find that, for $r \geq 1$,

$$1 - G_n(r) \leq 2 \left[1 + \frac{32}{3\sqrt{6\pi}} + \frac{4}{3\sqrt{3}} + \frac{\exp(40/9)}{8\sqrt{2e}} \right] \exp\left(-2\, r^2\right) . \tag{A1}$$

For $r < 1$, the expression on the right hand side is greater than one, so that (A1) is valid for all r.

Q.E.D.

ACKNOWLEDGEMENT

This research was supported by the Air Force Office of Scientific Research, Air Force Systems Command, USAF, under Grant AFOSR-76-3062.

REFERENCES

1. M. Loève, Probability Theory. New York: Van Nostrand, 1963.
2. W. Hoeffding, "Probability inequalities for sums of bounded random variables," J. Am. Stat. Assoc., vol. 58, pp. 13-30, 1963.
3. A. Dvoretzky, J. Kiefer, and J. Wolfowitz, "Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator," Ann. Math. Stat., vol. 27, pp. 642-669, 1956.
4. R. S. Singh, "On the rate for uniform strong consistency of empirical distributions of independent nonidentically distributed multivariate random variables," J. Multivariate Analysis, vol. 6, pp. 338-342, 1976.
5. L. P. Devroye, "A bound for the uniform deviation of empirical distribution functions," to appear in J. Multivariate Analysis.
6. G. L. Wise, A. P. Traganitis, and J. B. Thomas, "The estimation of a probability density function from measurements corrupted by Poisson noise," to appear in IEEE Trans. Inform. Theory.
7. W. Feller, An Introduction to Probability Theory and Its Applications, Vol. I. New York: Wiley, 1968, p. 54.

ROBUST IDENTIFICATION OF PARAMETERS OF LINEAR SYSTEMS

SANTANU BASU AND V. DAVID VANDELINDE
Department of Electrical Engineering
The Johns Hopkins University
Baltimore, Maryland 21218

ABSTRACT

We consider robust on-line estimation of crosscorrelations for the parameter identification of linear, stationary, SISO discrete systems. It is shown that the procedure provides consistent estimates with convergence in the square mean. When the measurement noise distribution has heavy tails, simulation studies have shown that the total r.m.s. error of these estimates are considerably lower than the error of the estimates obtained by the usual linear recursive scheme for estimating cross-correlations.

I. INTRODUCTION

The identification of linear, stationary, single-input-single-output, noisy systems is of considerable interest and has accordingly received much attention in recent years (Astrom and Eykhoff [1], Eykhoff [2] and Mehra and Lainiotis [3]). Among available techniques for identification, the most important ones from an engineering viewpoint are algorithms which perform on-line, and we shall restrict our discussion to such procedures. We note, however, that many of our arguments hold equally for off-line procedures.

In spite of the extensive studies in identification, certain important aspects of the problem have not received adequate consideration. We would like specifically to draw attention to the modeling of the measurement noise with regard to the role such models play in the design and performance of identification algorithms. In general, there have been somewhat extreme approaches to modeling the output perturbances. As an example, at one end of the scale is the maximum likelihood method which assumes for the measurement noise a known Gaussian distribution. If the actual noise has departures from this assumed model, there can be serious deterioration in the performance of the estimation scheme such as inconsistency of the estimates. At the other extreme is the method of crosscorrelations, which requires only that the input sequence and output noise be stationary and mutually uncorrelated. No knowledge of the distribution of the output noise is assumed or required. One might expect that such general assumptions would make the crosscorrelations method an attractive technique for identification and this indeed is the case. Comparative evaluations of on-line methods (Isermann et al [4], Saridis [5] and Sinha and Sen [6]) indicate clearly that it is one of the most efficient procedures available for the identification of general linear processes. However, the customary procedure for estimating crosscorrelations does have an inherent shortcoming which stems from the wide generality allowed the measurement noise: the recursive estimation scheme (as in [4]), though always consistent, is affected considerably, in terms of the variance of the estimates and rate of convergence, by the distribution of the additive output noise. Thus, in general, the performance of an identification method depends heavily on how close the actual noise is to the model assumed in the design of the algorithm. In this paper we introduce a new noise model and develop an identification procedure whose performance is less sensitive to the actual distribution of the output noise.

Firstly, since it seems realistic to assume some, but not complete, knowledge of the operating environment, we find it reasonable to use a noise model in which the output noise distribution is not completely specified (as a Gaussian process, for example), nor left completely unspecified (as in the crosscorrelation method). Secondly, we estimate the crosscorrelations of a linear stationary system by a new recursive scheme which utilizes our partial information about the output noise and whose performance is more uniform over different noise distributions. That is to say, we propose a robust estimator of crosscorrelations.

In Section II, we state the problem and provide a brief review of the ordinary crosscorrelations method along with its shortcomings. In Section III, we introduce the proposed algorithm designed to desensitize the effect of measurement noise and show this recursive scheme to be consistent in the square mean for binary inputs. In Section IV, we present simulation results which illustrate the robust nature of our algorithm. Comments and conclusions are given in Section V.

II. PROBLEM STATEMENT & REVIEW OF CROSSCORRELATIONS METHOD

Consider the problem of identifying the parameters $(a_1,\ldots,a_\ell,b_1,\ldots,b_\ell)$ of a linear, single input-single-output, stationary system modeled in discrete time as

$$\begin{aligned} y_n &= a_1 y_{n-1} - a_2 y_{n-2} - \ldots - a_1 y_{n-\ell} + b_1 u_{n-1} + \ldots + b_1 u_{n-\ell} \\ z_n &= y_n + \omega_n \end{aligned} \tag{1}$$

under the assumptions:

(i) the order, ℓ, of the system is known, and the system is completely controllable and observable;

(ii) $\{u_n\}$ and $\{\omega_n\}$ are stationary white noise sequences which are mutually uncorrelated;

(iii) the distribution of $\{\omega_n\}$ is given by

$$F(\omega) = (1-\varepsilon)K(\omega) + \varepsilon C(\omega) \qquad 0 \leq \varepsilon < 1 \tag{2}$$

where $K(\cdot)$ is completely specified and C is unknown but belongs to $\mathscr{S}$, the class of symmetric distributions with finite variance. ε may or may not be known; it is typically < 0.2.

The usual (and effective) procedure has been to consider the identification problem under assumptions (i) and (ii) only.

Since $\{u_n\}$ is stationary, its autocorrelation is given by

$$\begin{aligned} R_{uu}(m) &= E[u_n u_{n-m}] \\ &= \lim_{n\to\infty} \frac{1}{n+1} \sum_{i=0}^{n} u_i u_{i-m} \end{aligned} \tag{3}$$

and the crosscorrelation between the input $\{u_n\}$ and output $\{z_n\}$ is

$$\begin{aligned} R_{zu}(m) &= E[z_n u_{n-m}] \\ &= \lim_{n\to\infty} \frac{1}{n+1} \sum_{i=0}^{n} z_i u_{i-m} \end{aligned} \tag{4}$$

note that

$$\begin{aligned} R_{zu}(m) &= E[z_n u_{n-m}] = E[y_n u_{n-m}] + E[\omega_n u_{n-m}] \\ &= E[y_n u_{n-m}] = R_{yu}(m) \end{aligned} \tag{5}$$

The convolution equation

$$R_{yu}(m) = \sum_{i=0}^{\infty} h(i)\, R_{uu}(m-i) \tag{6}$$

relates the input-output through the impulse response, $h(i)$, of the system. Using an input sequence $\{u_n\}$ of known autocorrelations, estimates of $h(i)$ may be computed from R_{uu} and estimates of $R_{yu}(=R_{zu})$. In particular, if $\{u_n\}$ is white, we have

$$R_{uu}(m) = R_{uu}(0)\ \delta_m \quad \text{where} \quad \delta_m = \begin{cases} 1 & m=0 \\ 0 & m\neq 0 \end{cases}$$

so that

$$\hat{h}(m) = \frac{\hat{R}_{yu}(m)}{R_{uu}(0)} = \frac{\hat{R}_{zu}(m)}{R_{uu}(0)} \tag{7}$$

Using the $\{\hat{h}(i)\}$, the estimated response $\hat{y}(j)$ may be computed as in [4] by

$$\hat{y}(j) = \sum_{i=0}^{\infty} \hat{h}(j-i)u(i) \tag{8}$$

Then, using (1), we can write

$$\begin{bmatrix} \hat{y}(1) \\ \\ \\ \\ \\ \\ \hat{y}(j) \end{bmatrix} = \begin{bmatrix} 0 & & 0 & u(0) & 0 & (0)..0 \\ -\hat{y}(1) & & 0 & u(1) & u(0) & 0...0 \\ -\hat{y}(2) & -\hat{y}(1) & 0 & . & . & . \\ . & & . & . & . & . \\ . & & . & . & . & . \\ . & & . & . & . & . \\ -\hat{y}(j-1) & -\hat{y}(j-2) & -\hat{y}(j-\ell) & u(j-1) & & u(j-\ell) \end{bmatrix} \begin{bmatrix} \hat{a}_1 \\ \\ \hat{a}_n \\ \\ \hat{b}_1 \\ \\ \hat{b}_n \end{bmatrix} \tag{9}$$

$$\hat{y} \quad = \quad M \quad \hat{\theta}$$

The parameter vector $\hat{\theta}=(\hat{a}_1,\ldots,\hat{a}_\ell,\hat{b}_1,\ldots,\hat{b}_\ell)$ is then estimated by a least squares computation

$$\hat{\theta} = [M^T M]^{-1} M^T \hat{y} \tag{10}$$

Equations (8)-(10) are especially simple for a step input, i.e. $u(0) = u(1)=\ldots=u(j)=1$. The crosscorrelation $R_{zu}(m)$ is typically estimated on-line by writing (4) in recursive form ([4]):

$$\hat{R}_{zu}(m,n) = \hat{R}_{zu}(m,n-1) - \frac{1}{n+1}\{\hat{R}_{zu}(m,n-1)-z_n u_{n-m}\} \tag{11}$$

For symmetrically distributed $\{u_n\}$, (11) can be shown to be a stochastic approximation algorithm which converges in the square mean to the true values $\{R_{zu}(m)\}$. However, the shortcoming of this procedure is evident on a closer examination of (11). Considering the term $(\hat{R}_{zu}(m,n-1)-z_n u_{n-m})$, it is easy to see that the variance of $\hat{R}_{zu}$ and the rate of convergence of the algorithm depend on the distribution of $\{z_n u_{n-m}\}$. This might be expected naturally since it is the expected value of $\{z_n u_{n-m}\}$ that we are estimating. The distribution of $\{z_n u_{n-m}\}$ in turn is a function of the distribution of ω_n. Thus, algorithm (11) provides consistent estimates of R_{zu} but the variance of these estimates and rate of convergence of the algorithm depend on the nature of ω_n, which can be quite general under assumption (ii). We need, therefore, to stabilize the estimation scheme for R_{zu} against the effect of ω_n and we do so in the next section.

III. A CONSISTENT ROBUST ESTIMATOR FOR R_{zu}

Since the shortcoming of the linear recursive scheme (11) stems from the generality of $\{\omega_n\}$, we incorporate assumption (iii), which implies that ω_n is partially specified. We next propose the following non-linear recursive scheme to reduce the effect of ω_n on the estimation procedure, which is thus made robust:

$$\hat{R}_{zu}(m,n) = \hat{R}_{zu}(m,n-1) - \frac{1}{n+1}\, g\{R_{zu}(m,n-1) - a_n u_{n-m}\} \tag{12}$$

where $g(\cdot)$ is an odd, monotonically increasing, continuous and bounded function defined by

$$\begin{aligned} g(x) &= x & |x| &\leq d_1 \\ &= d_1 \operatorname{sgn}(x) & |x| &\geq d_1 \end{aligned} \tag{13}$$

where d_1 is a constant chosen as a function of the parameters of $K(\omega)$, the specified part of the distribution of ω_k (see (2)). $g(\cdot)$ is the 'soft limiter' used by Huber [7] in the robust estimation of a location parameter. We might add the observation that (12) seems a natural - but not necessarily the only - way to stabilize the estimation of R_{zu} against the distribution of ω_n, since at any stage k, at most $A_k . d_1$ is either added to or subtracted from the previous estimate. There are no bounds for this term in the common procedure (11). We mention here that non-linear estimates of autocorrelations have been used earlier by Huzii [8,9] for stationary Gaussian processes and by Rodemich [10] for more general stationary processes. However, the motivation of both these investigators was a simplification of the computational effort and not the desensitization of the estimation technique to different noise processes. Some investigations of the robust parameter estimation problem have been initiated, in somewhat different formulations than the one here, by Nasburg and Kashyap [11], Denby and Martin [12] and Martin and Jong [13].

For notational convenience, rewrite (12) as

$$x_n = x_{n-1} - A_n\, g(x_{n-1} - z_n u_{n-m}) \tag{14}$$

where $x_n = \hat{R}_{zu}(m,n)$

$$A_n = \frac{1}{n+1}$$

m can be suppressed since it is fixed for each set of iterations.

Also set $R_{zu}(m) = h(m)\, R_{uu}(0) = h_m R = X$ (15)

Before we consider the main result of this paper, we state a couple of lemmas (without proof, for the sake of brevity).

Lemma 1 If $\{u_n\}$ is white and symmetrically distributed, the distribution of $(z_n u_{n-m} | u_{n-m})$ is symmetric around its mean $h_m u^2_{n-m}$.

Lemma 2 If $\{u_n\}$ is a white binary sequence, then $E_u\{g(h_m R - z_n u_{n-m})\}=0$.

We are now in a position to state and prove the main result:

Theorem

For the identification of system (1) under the assumptions (i)-(iii) and $\{u_n\}$ a white binary sequence, the scheme (12) (or (14)) provides consistent estimates of the crosscorrelations $\{R_{zu}(m)\}$, with convergence in the square mean:

i.e. $\hat{R}_{zu}(m,n) \to R_{zu}(m)$ in the square mean.

Proof (We have essentially followed the method of Robbins and Monro [14].) Recalling (14),

$$x_n = x_{n-1} - A_n\, g(x_{n-1} - z_n u_{n-m}) \qquad A_n = \frac{1}{n+1}$$

$$\Sigma A_n = \infty$$

$$\Sigma A_n^2 < \infty$$

Remembering $X = h_m R = R_{zu}(m)$, we set

$$x_n - X = (x_{n-1} - X) - A_n\, g(x_{n-1} - z_n u_{n-m}) \tag{16}$$

Squaring both sides,

$$\begin{aligned}(x_n - X)^2 = {} & (x_{n-1} - X)^2 + A_n^2\{g(x_{n-1} - z_n u_{n-m})\}^2 \\ & -2\, A_n (x_{n-1} - X)\ g(x_{n-1} - z_n u_{n-m})\end{aligned} \tag{17}$$

Taking expectations w.r.t. x_{n-1} ,

$$\begin{aligned}E\{(x_n - X)^2 | x_{n-1}\} = {} & (x_{n-1} - X)^2 + A_n^2\ E[\{g(x_{n-1} - z_n u_{n-m})\}^2 | x_{n-1}] \\ & - 2\, A_n (x_{n-1} - X)\ E\{g(x_{n-1} - z_n u_{n-m}) | x_{n-1}\}\end{aligned} \tag{18}$$

Set $E\{g(x_{n-1} - z_n\, u_{n-m}) | x_{n-1}\} = \beta(x_{n-1})$

There are three cases to be considered:

(a) $x_{n-1} = X$: $\beta(x_{n-1}) = 0$ by lemma 2. (19)

(b) $x_{n-1} > X$: Point by point $(x_{n-1} - z_n u_{n-m}) > (X - z_n u_{n-m})$, and since $g(\cdot)$ is an odd, continuous, monotonically increasing function, $\beta(x_{n-1}) > 0$. (20)

(c) $x_{n-1} < X$: Point by point $(x_{n-1} - z_n u_{n-m}) < (X - z_n u_{n-m})$ and for reasoning similar to that in (b) above, $\beta(x_{n-1}) < 0$. (21)

Setting $E(x_n - X)^2 = E_n$, and using (18), we obtain

$$\begin{aligned}E_n = {} & E_{n-1} + A_n^2\ E\{E[\{g(x_{n-1} - z_n u_{n-m})\}^2 | x_{n-1}]\} \\ & - 2\, A_n\ E[(x_{n-1} - X)\,\beta(x_{n-1})]\end{aligned} \tag{22}$$

Setting $E\{E[\{g(x_{n-1} - z_n u_{n-m})\}^2 | x_{n-1}]\} = f_n$

and $E[(x_{n-1} - X)\ \beta(x_{n-1})] = e_n$

it is evident that $0 \leq f_n < \infty$ and using (19), (20) and (21), $e_n \geq 0$ (See [14]). Thus

$$E_n = E_{n-1} + A_n^2 f_n - 2 A_n e_n \tag{23}$$

Summing over (22), we get

$$E_n = E_1 + \sum_{j=2}^{n} A_j^2 f_j - 2 \sum_{j=2}^{n} A_j e_j \tag{24}$$

Since $E_n \geq 0$,

$$\sum_{j=2}^{n} A_j e_j \leq \frac{1}{2} (E_1 + \sum_{j=2}^{\infty} A_j^2 f_j) < \infty \tag{25}$$

So, the positive term series $\sum_{j=2}^{\infty} A_j e_j$ converges.

Thus, $\lim_{n\to\infty} E_n = E_1 + \sum_{2}^{\infty} A_j^2 f_j - 2 \sum_{2}^{\infty} A_j e_j = E$ exists, and $E \geq 0$.

Since $A_n = \frac{1}{n+1}$, we have, from [14], $\lim_{n\to\infty} E_n = E = 0$.

i.e. $\lim_{n\to\infty} E(x_n - X)^2 = 0$, or, $\hat{R}_{zu}(m,n) \to R_{zu}(m)$ in square mean.

In the next section we present some simulation results to illustrate the performance of the new recursive scheme (12) as compared to (11) for various distributions of ω_n.

IV. SIMULATION STUDIES

To examine the performance of the new estimator (12) for R_{zu} as compared to the common estimator (11), we have carried out digital computer simulations for the identification of a second order non-minimum phase linear process used as a test case in [4]. We have used various distributions for $\{\omega_n\}$ to ascertain, in particular, the effect of the measurement noise on the identification schemes.

The simulated second order non-minimum phase process that was identified was

$$y_n = -a_1 y_{n-1} - a_2 y_{n-1} + b_1 u_{n-1} + b_2 u_{n-2}$$

$$z_n = y_n + \omega_n$$

where $a_1 = -1.425 \quad a_2 = 0.496$

$b_1 = -0.102 \quad b_2 = 0.173$

The input $\{u_n\}$ was a white binary sequence of amplitude +0.5 or -0.5. Thus, the autocorrelation $R_{uu}(0) = (0.5)^2 = 0.25 = E(u_n^2)$.

For the corrupting noise ω_n, we chose distributions $F(\cdot)$ of the contaminated Gaussian class, where the known part $K(\cdot)$ of ω_n is a specified Gaussian distribution. Such a model has been widely applied since it seems reasonable from a practical viewpoint. The contaminations (i.e. the unknown part of $F(\omega)$) are either Gaussian with larger variances or Laplacian (double exponential) of different variances. We used 20 different distributions for F, using various values of ε and different variances for $K(\cdot)$ and $C(\cdot)$.

We report here, for brevity, the results for the following 3 distributions (with means and variances in parentheses):

a. $\omega_n \sim G(0,1)$ (i.e $F(\cdot)$ is completely known)

b. $\omega_n \sim 0.85\ G(0,1) + 0.15\ L(0,10)$

c. $\omega_n \sim 0.97\ G(0,1) + 0.03\ L(0,100)$

The following estimators were used for the on-line estimation of $R_{zu}(m)$:

i. Linear $d_1=\infty$ (in equation (13) defining $g(\cdot)$)
ii. Limiter 1 $d_1=2\sigma$
iii. Limiter 2 $d_1=1.5\sigma$
iv. Limiter 3 $d_1=\sigma$

where σ= standard deviation of the known distribution $K(\cdot)$.

(in our examples $K(\cdot) = G(0,1)$)

The identification algorithms for each estimator were run for 3000 stages of the process, and parameter estimates were computed every 300 stages using the least squares scheme in Section II. The number of impulse response values, j, in eq. (9) was taken as 4.

The error was computed as the normalized mean squared error:

$$\text{Error } \delta = \frac{\|\Delta\theta_i\|}{\|\theta_i\|} = \left[\frac{\sum\limits_{i=1}^{4}(\theta_i-\hat{\theta}_i)^2}{\sum\limits_{i=1}^{4}\theta_i^2}\right]^{1/2} \tag{27}$$

where θ_i = the true parameters $\{a_i\}$, $\{b_i\}$
$\hat{\theta}_i$ = the estimates $\{\hat{a}_i\}$, $\{\hat{b}_i\}$

To obtain better comparisons of the different estimators, 30 Monte Carlo runs of each identification iteration (of 3000 stages) were done and the total r.m.s. error was computed for each estimator. If k refers to each Monte Carlo run, then total r.m.s. error

$$\delta_r = \left[\frac{1}{30}\sum_{k=1}^{30}(\delta(k))^2\right]^{1/2} \tag{28}$$

The results of the simulation are shown in Figures 1,2 and 3. When the corrupting noise has low variance (i.e. Fig. 1, $\omega_n \sim G(0,1)$) the non-linear estimators of (12) perform only as well or somewhat worse than the linear estimator. However, that is a small price to pay for the marked improvement shown by the non-linear estimators over the linear estimator in the cases of high contamination (Fig. 2, $\omega_n \sim .85\ G(0,1)+.15\ L(0,10)$) and high contaminating variance (Fig. 3, $\omega_n \sim .97\ G(0,1)+.03\ L(0,100)$). It is clear that the new estimator (12) is less sensitive to the distribution of the measurement noise and thus has more uniform - or, robust - performance than the normal estimator against varying distributions for ω_n.

V. CONCLUSIONS AND COMMENTS

We have presented here an approach to the design of an identification algorithm whose performance is less sensitive to the measurement noise. We have proved consistence of our proposed estimator and illustrated its robustness by simulation studies. A few additional observations remain to be made.

Simulation studies have also been carried out for the other two test cases used in [4] and results similar to the ones reported here in Section IV have been obtained. We have found that the robust nature of the algorithm is also exhibited for redescending g functions - i.e, those that return to and stay at zero beyond a point $\pm d_2 (|d_2| > |d_1|$, see eq. (13) for $g(\cdot)$). However, we have not been able to prove convergence in this case.

The questions of immediate interest from the performance viewpoint are the choice of d_1 (the saturation point for $g(\cdot)$) and the number of impulse response values, j, used for the least squares computation of $\hat{\theta}$ in eq. (10). Theoretically, the asymptotic distribution of these sequential estimates is of primary importance. We have undertaken investigations in this direction as well as the extension of our results to include more general inputs $\{u_n\}$.

REFERENCES

[1] K. J. Åström and P. Eykhoff (1971), "System identification - a survey", Automatica, Vol. 7, 123-162.

[2] P. Eykhoff (1974), "System identification - parameter and state estimation", John Wiley & Sons, New York.

[3] R. Mehra and D. Lainiotis (1976), eds., "System identification:Advances and Case studies", Academic Press, New York.

[4] R. Isermann, U. Baur, W. Bamberger, P. Kneppo and H. Siebert, (1974), "Comparison of six on-line identification and parameter estimation methods", Automatica, Vol. 10, 81-103.

[5] G. N. Saridis (1974), "Comparison of six on-line identification algorithms", Automatica, Vol. 10, 69-79.

[6] N. K. Sinha and A. Sen (1975), "Critical evaluation of on-line identification methods", Proc. IEE, Vol. 122, 1153-1158.

[7] P. J. Huber (1964), "Robust estimation of a location parameter", Annals Math. Stat., Vol. 35, 72-100.

[8] M. Huzii (1963), "On a simplified method of the estimation of the correlogram for a stationary Gaussian process", (Tokyo) Inst. of Stat. Math. Annals, Vol. 14, 259-268.

[9] M. Huzii (1970), "On the variance of a simplified estimate of a correlogram", (Tokyo) Inst. of Stat. Math. Annals, Vol. 22, 527-534.

[10] E. R. Rodemich (1966), "Spectral estimates using non-linear functions", Annals of Math. Stat., Vol. 37, 1237-1256.

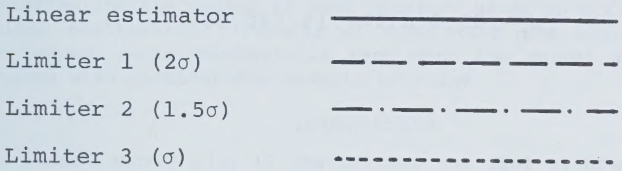
[11] R. E. Nasburg and R. L. Kashyap (1975), "Robust parameter estimation in dynamic systems", Proc. JHU Conference on Infor. Sciences and Systems.

[12] L. Denby and R. D. Martin (1975), "Robust fitting of auto-regressive models", Annual meeting of ASA, Atlanta, Georgia.

[13] R. D. Martin and J. Jong (1975), "Asymptotic properties of M-estimates for the transition parameter of a 1st order Markov model", private communication, submitted for publication.

[14] H. Robbins and S. Monro (1951), "A stochastic approximation method", Annals Math. Stat., Vol. 22, 400-407.

** This research was supported by U.S. Army Research Office grant number DAAG 29-77-G-0093.



Legend for Figures 1-3

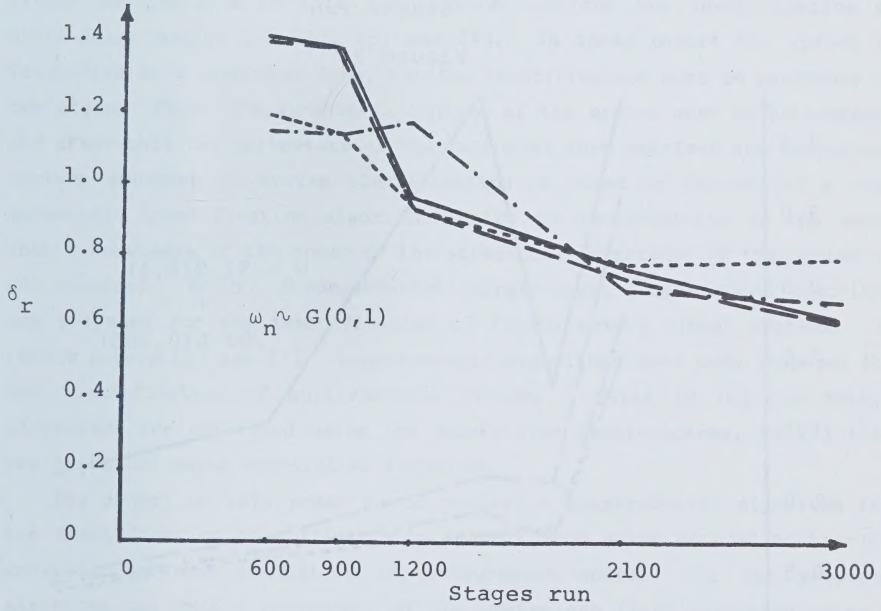


Figure 1

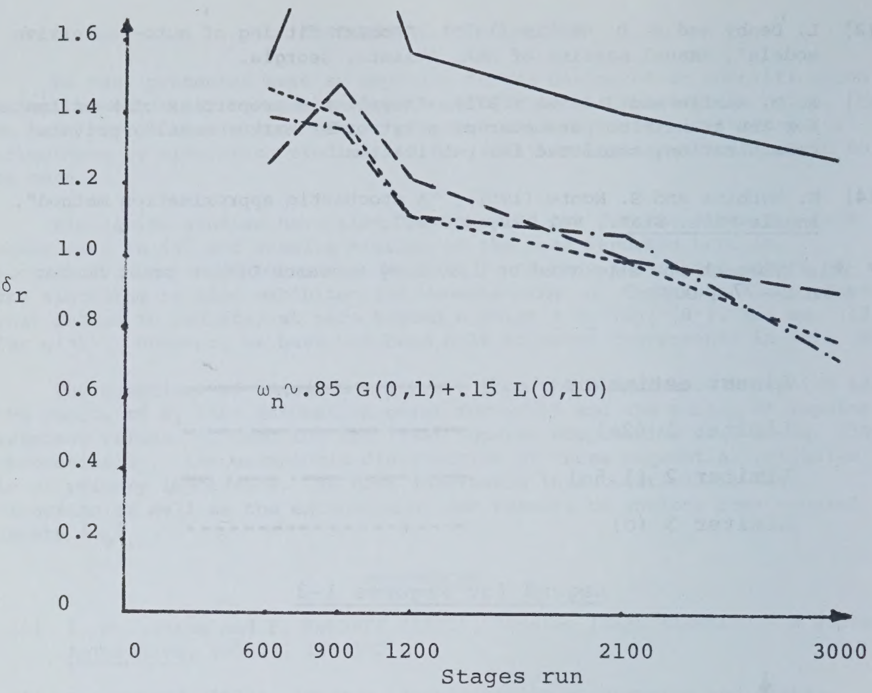


Figure 2

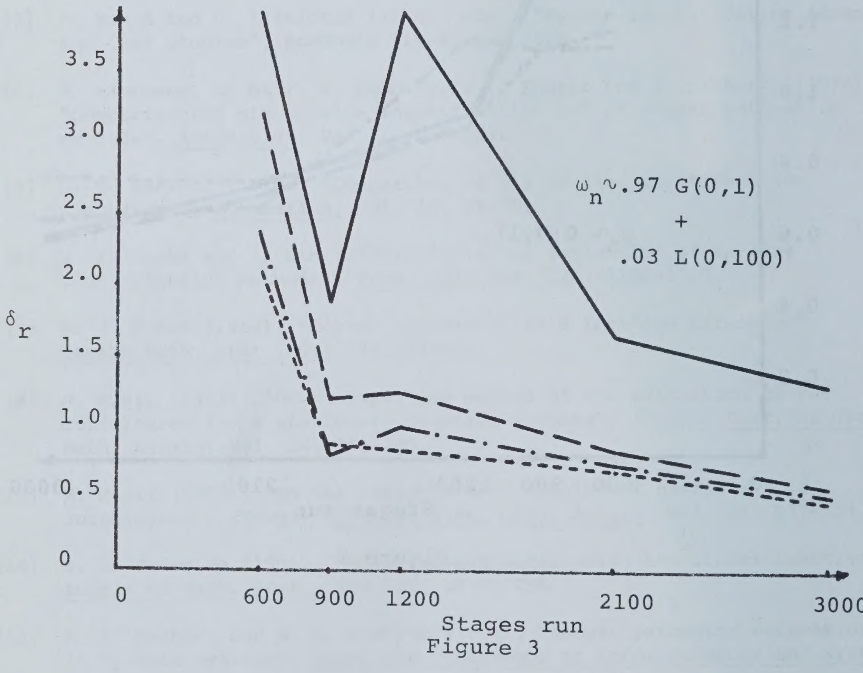


Figure 3

A NONPARAMETRIC STOCHASTIC APPROXIMATION ALGORITHM FOR ON-LINE IDENTIFICATION OF MULTIVARIABLE SYSTEMS

H. EL-SHERIEF and N.K. SINHA
Department of Electrical Engineering
McMaster University, Hamilton, Canada

ABSTRACT

A nonparametric algorithm is proposed for on-line identification of linear multivariable discrete-time systems from the measurements of input-output data contaminated with noise. It is based on the estimation of the Markov parameters of the system from the measured data using a nonparametric normalized stochastic approximation algorithm. The state-space representation for the system is then obtained using an efficient algorithm for minimal realization. Results of simulation are included indicating that the scheme works successfully even when the output measurements are contaminated with considerable amounts of noise.

INTRODUCTION

In a recent survey [1], it was pointed out that although a great deal of work has been done on the identification of single-input single-output systems, relatively little has been published in the identification of multivariable systems from input-output data contaminated with noise. The latest publications in this field which consider the identification of state-space models are [2], [3] and [4]. In these papers the system is identified in a canonical form, but the identification must be performed in two steps. First the structural indices of the system must be determined, and after this the parameters of the canonical form matrices are estimated. Another approach to system identification is based on the use of a nonparametric identification algorithm, which is nonparametric in the sense that a knowledge of the order or the structural invariants of the system is not required. In [5], a nonparametric single-input single-output algorithm was proposed for the identification of finite memory linear systems. In recent papers [6] and [7] nonparametric algorithms have been proposed for the identification of multivariable systems. While in [6] the Markov parameters are estimated using the generalized least-squares, in [7] they are estimated using correlation technique.

The object of this paper is to present a nonparametric algorithm for the identification of multivariable systems from noisy data without prior knowledge of the statistics of measurement noise. In the proposed algorithm the Markov parameters of the system are first estimated using a nonparametric normalized stochastic approximation algorithm proposed in [8], then from the estimates of these parameters, the state-space models are obtained directly using an efficient algorithm for minimal realization proposed in [9].

STATEMENT OF THE PROBLEM

Consider a linear, time invariant, discrete-time system which is described by the following set of equations.

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + v(k) \end{aligned} \tag{1}$$

where x(k) is the n-dimensional state vector, y(k) is the m-dimensional output measurement vector, u(k) is the r-dimensional input vector and v(k) is the m-dimensional output noise vector (not directly measured). A, B and C are system matrices of proper dimensions.

Our aim is to identify the matrices A, B and C of system (1) through subsequent observations of the variables u(k) and y(k) without knowing the statistics of the measurement noise and the structural indices of the system.

DERIVATION OF THE PROPOSED ALGORITHM

Assuming zero initial conditions, the state of system (1) at time k is given by

$$x(k) = \sum_{i=0}^{\infty} A^i B\, u(k-i-1)$$

which can be rewritten as

$$x(k) = \sum_{i=0}^{L-1} A^i B\, u(k-i-1) + \sum_{i=L}^{\infty} A^i B\, u(k-i-1) \tag{2}$$

where L is called the identification interval.

Using equation (2), the outputs of system (1) can be written as

$$y(k+L) = \sum_{i=0}^{L-1} J_i\, u(k+L-i-1) + \varepsilon(k-1) + v(k+L) \tag{3}$$

where the matrices J_0, J_1, J_2, ... are called the Markov parameters of the system and are defined as,

$$J_i = C A^i B$$

and

$$\varepsilon(k-1) = \sum_{i=L}^{\infty} J_i\, u(k+L-i-1)$$

The value of the identification interval L, in equation (3), depends on the degree of approximation obtained by truncating the series of Markov parameters. As it was shown in [8] increasing the value of L will give better estimates of the Markov parameters but on the other hand will increase the computation effort.

Our objective now is to identify the Markov parameters of equation (3) from the input output data without the knowledge of the statistics of the noise contaminating the measurements.

Defining the following (Lr)-dimensional auxiliary vectors

$$\theta_i^T = [J_{0,i} \;\; J_{1,i} \; \dots \; J_{L-1,i}] \tag{4}$$

$$U^T(k+L-1) = [u^T(k+L-1) \; u^T(k+L-2) \; \dots \; u^T(k)]$$

where J_ℓ, $\ell = 0, 1, \dots, L-1$ has been partitioned as

$$J_\ell = \begin{bmatrix} J_{\ell,1} \\ J_{\ell,2} \\ \vdots \\ J_{\ell,m} \end{bmatrix}$$

Then the outputs of system (3) can be rewritten as

$$y_i(k+L) = U^T(k+L-1)\;\theta_i + e_i \tag{5}$$

where

$$e_i = \varepsilon_i(k-1) + v_i(k+L) \qquad i = 1, 2, \dots, m$$

System (3) has been decomposed into m single output subsystems (equation (5)), the parameter vectors θ_i's of each subsystem will be now estimated independently.

Defining $\hat{\theta}_i$ as the identification estimates of the unknown parameter vectors θ_i, $i = 1, 2, \dots, m$ of equation (5). Then the estimates $\hat{\theta}_i$, $i = 1, 2, \dots, m$ are obtained recursively by means of the following set of normalized stochastic approximation algorithms [8],

$$\hat{\theta}_i(k+L) = \hat{\theta}_i(k-1) + \nu(k)\,\frac{U(k+L-1)}{||U(k+L-1)||^2}\,[y_i(k+L) - U^T(k+L-1)\;\hat{\theta}_i(k-1)]$$
$$i = 1,2,\dots,m \qquad k = 1,2,\dots \tag{6}$$

where the sequence $\nu(k)$ and the initial estimates satisfy the following conditions:

$$\lim_{i\to\infty} \nu(i) = 0, \quad \sum_{i=1}^{\infty} \nu(i) = \infty, \quad \sum_{i=1}^{\infty} \nu^2(i) < \infty$$

and

$$E\{||\hat{\theta}_i(0)||^2\} < \infty \qquad i = 1, 2, \dots, m$$

After identifying the unknown parameter vectors θ_i's, the Markov parameters of the system are obtained from relation (4). The system matrices A, B and C can be obtained from the resulting Hankel matrix for the system using an efficient algorithm for minimal realization proposed in

[9].

SIMULATION RESULTS

The proposed algorithm was applied to the identification of a simulated third-order system with one input and two outputs described by the following equations

$$x(k+1) = \begin{bmatrix} 0 & 0 & 0.025 \\ 1 & 0 & -0.1 \\ 0 & 1 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u(k)$$

$$y(k) = \begin{bmatrix} 0.12 & 0.36 & 0.14 \\ 0.20 & 0.29 & 0.559 \end{bmatrix} x(k) + v(k)$$

where, $x(k)$, $u(k)$, $y(k)$, $v(k)$ are the state vector, the scalar input, the measurement output vector and the output noise vector respectively. The scalar input $u(k)$ was a zero-mean white noise sequence with unit variance and each of the two outputs was contaminated with an additive white noise sequence with variance adjusted to make the noise-to-signal ratio at each output equal to 66.2%.

The Markov parameters of the above system are easily found as

$$J_0 = \begin{bmatrix} 0.12 \\ 0.20 \end{bmatrix}, \quad J_1 = \begin{bmatrix} 0.36 \\ 0.29 \end{bmatrix}, \quad J_2 = \begin{bmatrix} 0.14 \\ 0.559 \end{bmatrix}, \quad J_3 = \begin{bmatrix} 0.107 \\ 0.535 \end{bmatrix}, \quad J_4 = \begin{bmatrix} .102 \\ .486 \end{bmatrix}$$

The Markov parameters of the above system were estimated, using algorithm (6) with $L = 10$, from 900 samples of the input-output data. The final estimates of the first five Markov parameters are obtained as

$$J_0 = \begin{bmatrix} 0.113 \\ 0.204 \end{bmatrix}, \quad J_1 = \begin{bmatrix} 0.361 \\ 0.310 \end{bmatrix}, \quad J_2 = \begin{bmatrix} 0.141 \\ 0.586 \end{bmatrix}, \quad J_3 = \begin{bmatrix} 0.112 \\ 0.537 \end{bmatrix}, \quad J_4 = \begin{bmatrix} 0.102 \\ 0.482 \end{bmatrix}$$

To show the rate of convergence of the proposed algorithm the error norm of the Markov parameters has been plotted against the number of samples and is shown in the figure. The error norm used is defined as

$$\text{Error norm} = \frac{||\hat{\theta}(k) - \theta||^2}{||\hat{\theta}(0) - \theta||^2}$$

where θ is a vector formulated from the Markov parameters, that is;

$$\theta = [0.12 \quad 0.2 \quad 0.36 \quad 0.29 \quad 0.14 \quad 0.559 \quad 0.107 \quad 0.535 \quad 0.102 \quad 0.486]$$

and $\hat{\theta}(0)$ is the initial estimate of θ which was taken as zero.

From the estimated Markov parameters, assuming that the order of the system is known, the following matrices are obtained using the minimal realization algorithm proposed in [9]

$$\hat{A} = \begin{bmatrix} 0 & 0 & 0.015 \\ 1 & 0 & -0.089 \\ 0 & 1 & 1.020 \end{bmatrix}$$

$$\hat{B} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\hat{C} = \begin{bmatrix} 0.113 & 0.361 & 0.141 \\ 0.204 & 0.310 & 0.586 \end{bmatrix}$$

It can be seen from the results that the proposed algorithm gives unbiased and consistent estimates of the system parameters.

CONCLUSIONS

In this paper a nonparametric algorithm has been presented for the identification of multivariable systems from noisy measurements. The main advantage of the proposed algorithm is that it requires simple computations as it uses the stochastic approximation method. Furthermore, since the estimates can be improved in a recursive manner, the algorithm can be used for on-line identification.

Results of simulation indicate that the proposed algorithm gives good estimates in the presence of noise.

REFERENCES

[1] N.S. Rajbman and N.K. Sinha, "Identification of multivariable systems: A critical survey", Proc. 14 Annual Allerton Conference on Circuits and Systems, 1976, pp. 998-1006.

[2] R. Guidorzi, "Canonical structures in the identification of multivariable systems", Automatica, vol. 11, 1975, pp. 361-374.

[3] A. Tse and H.L. Weinert, "Structural determination and parameter identification for multivariable stochastic linear systems", IEEE Trans. on Automatic Control, vol. AC-20, 1975, pp. 603-613.

[4] N.K. Sinha and Y.H. Kwong, "Recursive identification of the parameters of a multivariable system", 4th IFAC Symp. on Multivariable Technological Systems (Fredericton, N.B., Canada), 1977, pp. 323-332.

[5] Y. Bar-Shalom and S.C. Schwartz, "A nonparametric recursive identification algorithm for finite memory linear systems", Proc. 7th Annual Allerton Conf. on Circuit and System Theory, 1969, pp. 523-532.

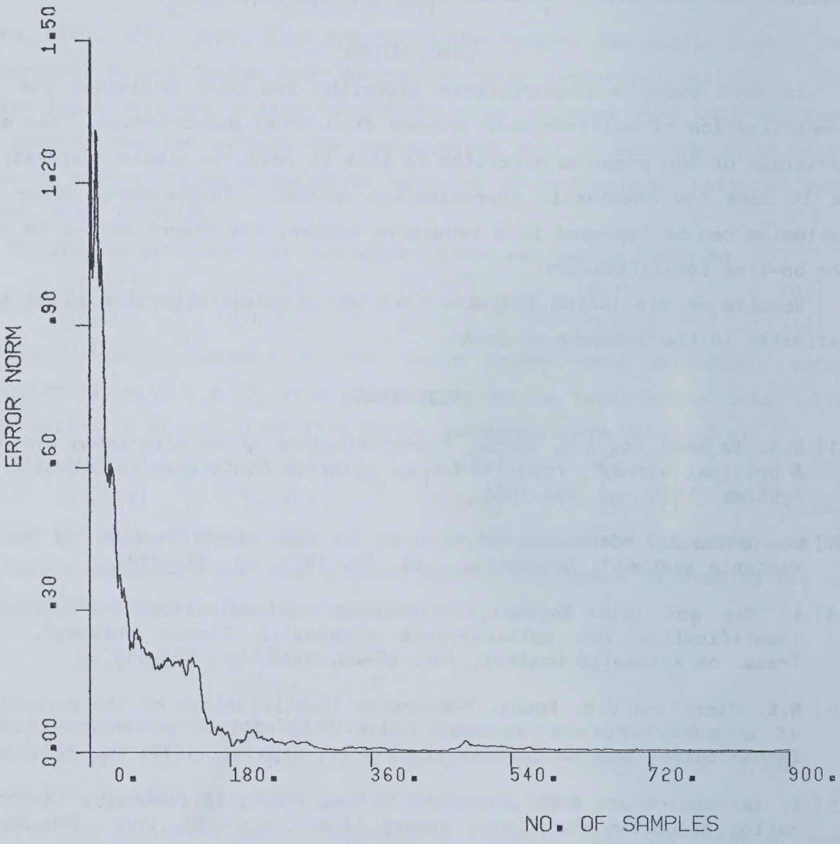
[6] N.K. Sinha and A. Sen, "On line identification of multivariable systems", Preprints IV IFAC Symp. on System Identification (U.S.S.R.), Sept. 1976.

[7] N.K. Sinha, A.K. Mahalanabis and H. El-Sherief, "A nonparametric approach to the identification of linear multivariable systems", to

appear in International Journal of System Science.

[8] H. El-Sherief and N.K. Sinha, "Stochastic approximation for the identification of multivariable systems", Faculty of Engineering, McMaster University, Hamilton, Canada, Report SOC-179, 1977.

[9] P. Rózsa and N.K. Sinha, "Efficient algorithm for irreducible realization of a rational matrix", International Journal of Control, vol. 20, November 1974, pp. 739-751.



Rate of convergence of the proposed algorithm

ON OPTIMUM LINEAR ESTIMATION OF DIGITAL SEQUENCES

ENZO DALLE MESE* and DINO GIULI**
* Department of Electrical Engineering, University of Pisa, Italy
**Institute of Electronics, University of Florence, Italy

In this paper we will consider the problem of linearly estimating a digital data sequence from indirect noisy measurements. The problem will be formulated and solved by means of Hilbert space methods and by utilizing the innovations of the observed sequence. Though we do not make any assumptions on the digital data sequence model (in particular we do not use a state space representation) the structure of the optimum linear estimator turns out to be a Kalman filter followed by a transversal filter with an infinite number of tap gains.Recursive formulae for computing the tap gains and the error covariance matrix are given. Finally, we apply the results obtained to the problem of eliminating the intersymbol interference in digital communications.

INTRODUCTION

The problem of linear estimation of digital sequences has undoubtedly a great practical importance as well as a theoretical one. In the field of communication engineering this topic is strictly connected with the elimination of intersymbol interference in data communication through a dispersive channel. The best linear solution to this problem is the one which Lawrence and Kaufman derived in 1971 [1] by means of a state space representation and the results of the Kalman filtering theory. However, it is known that this is not the optimum solution, i.e., with minimum error variance, and this is due mainly to the fact that a finite-dimensional model is used to solve an essentially infinite-dimensional problem.

In this paper, we shall discuss the problem of sequence estimation from indirect noisy measurements. This study, which in [2] was limited to the deterministic case, is now extended to cover the stochastic case. Assuming that all random processes considered are second order processes and that the measurement model is linear, we will find the linear estimate of the whole data sequence with minimum error variance. The difficulty connected with the infinite dimensionality of the problem is avoided by forcing the estimate to be recursive, so that at each step the problem is a finite-dimensional one. Moreover, the attribute of recursivity is very important if an on-line implementation of the estimator is desired.

The problem is solved by means of Hilbert space methods and by utilizing the concept of innovations of the observed sequence. This geometric approach was used by Mosca [3,4] to solve a very general linear regression problem, formulated in an abstract mathematical framework. His general results will be very useful in finding the solution to our problem.

Finally, we should like to point out that the same results can be obtained if the standard Kalman filtering theory is utilized with an appropriate state model. Instead, the approach followed in this paper does not make use of any state-space representation and so the significant results are obtained directly and very neatly. Moreover this approach has a very simple and somewhat suggestive geometric interpretation that allows us to unify seemingly different problems (see, for example [2]). These reasons, in our opinion, justify and encourage this approach.

PROBLEM STATEMENT

Suppose that we observe a random white data sequence, $\{u(k),k=1,2,...\}$, transformed according to a linear measurement system and inbedded in an additive white noise sequence $\{n_k',k=1,2,...\}$ uncorrelated with $\{u(k),k=1,2,...\}$, as schematically depicted in Fig. 1. Let $\{z_k',k=1,2,...\}$ be the observed sequence. We will assume that all the random variables we will be involved with are second order random variables and zero-mean.

Obviously, the second assumption does not lead to any loss of generality since we are interested in linear operations only.

Moreover, suppose that the causal linear measurement system has finite memory, say L, so that only the first L samples of its discretized impulse response $\{c(k),k=1,2,...\}$ are possibly not equal to zero. If we begin observations at time t=L so as to leave out any transient effect on the observed sequence, we have

$$z_k = \sum_{s=k}^{L-1+k} u(s)c(L+k-s)+n_k \quad , \quad k=1,2,... \tag{1}$$

where $z_k \triangleq z'_{k+L-1}$ and $n_k \triangleq n'_{k+L-1}$.

Let $\sigma_u^2(k)$ and $\sigma_n^2(k)$ be the variance of the random variables $u(k)$ and n_k respectively.

Within this framework the problem to be solved is the following: find a linear, recursive estimate of the random sequence $\{u(k),k=1,2,...\}$ from samples of a time series $\{z_k, k=1,2,...\}$, with minimum error variance.

Geometric formulation

Let us consider the linear space of all real- valued random variables defined on a probability space $(\Omega,\mathcal{A},\mathcal{P})$ [5] with finite second moment. This space, with the usual definition of the inner product, turns out to be a Hilbert space, say H. Let $\langle\cdot,\cdot\rangle$ and $\|\cdot\|$ represent the inner product and norm in H. Define H^k as the (k+L-1)-fold Cartesian product of H, i.e. the space of all random column vectors with (k+L-1) components. Let $u \in H^p$, with p possibly infinite, be the vector representing the random data sequence and C_k the deterministic row vector $[0,...,0,c(L),...,c(1),0,0,...]$, where the first component other than zero is the k-th. With these notations eq. (1) may be rewritten as follows

$$z_k = C_k u + n_k \quad , \quad k=1,2,... \tag{2}$$

where z_k, k=1,2,... is a sequence of random variables belonging to H. Let M_k be the column vector representing the cross-correlations between the components of u and z_k, i.e.

$$M_k \triangleq E(uz_k) = [\langle u,z_k\rangle] \tag{3}$$

having defined with $[\langle u,z_k\rangle]$ the column vector of the inner products between vectors u(i) and $z_k \in$ H

$$[\langle u,z_k\rangle] \triangleq [\langle u(1),z_k\rangle , \langle u(2),z_k\rangle,\dots]' \tag{4}$$

Notice that the components $M_k(i)$ of vector M_k with $i \geqslant L+K$ are all zero.

Now we are able to formulate our problem in a more suggestive way: find a recursive formula for $\hat{u}_{|t} \triangleq [\hat{u}_{|t}(1), \hat{u}_{|t}(2), \dots, \hat{u}_{|t}(p)]'$ $t=1,2,\dots$, where for each $i=1,2,\dots,p$, $\hat{u}_{|t}(i)$ is the minimum norm element in H, whose projections onto vectors $\{z_k, k=1,2,\dots, t\}$ are given by $\{M_k(i)\, \| z_k \|^{-1} \quad, k=1,2,\dots,t\}$. This is equivalent to saying that for each i and t $\hat{u}_{|t}(i)$ is a linear combination of $\{z_k, k=1,2,\dots,t\}$ and that $\| \hat{u}_{|t}(i)-u(i)\|$, the estimate error variance, is minimized.

SOLUTION BY INNOVATIONS

Let

$$\mathcal{S}_t \triangleq \mathrm{Span}\{z_i, i=1,2,\dots,t\} \tag{5}$$

be the linear manifold in H spanned by $\{z_i, i=1,2,\dots,t\}$,

$$\mathcal{P}_t(\cdot) \triangleq \mathrm{Project}(\cdot \mid \mathcal{S}_t)$$

be the orthogonal projection operator mapping H onto $\mathcal{S}_t$, and

$$\mathcal{J}_t(\cdot) \triangleq (I - \mathcal{P}_t)(\cdot)$$

be the orthogonal projection operator mapping H onto $\mathcal{S}_t^{\perp}$, the orthogonal complement of $\mathcal{S}_t$ with respect to H. I is the identity operator of H.

Now the minimum norm estimate, $\hat{u}_{|t} \in H^p$, which interpolates the cross-correlations $\{M_i, i=1,2,\dots,t\}$ can be computed through the following steps:

i) use the Gram-Schmidt procedure to orthonormalize the sequence of vectors $\{z_i\}$ to obtain

$$e_t \triangleq z_t - \mathcal{P}_{t-1}(z_t) = \mathcal{J}_{t-1}(z_t) \tag{6}$$

$$\nu_t \triangleq e_t / \| e_t \| ; \tag{7}$$

ii) determine the projections N_i of the unknown vector u onto the sequence of vectors $\{\nu_i\}$, by means of the knowledge of its projections onto the vectors $\{z_i\}$;

iii) compute the estimate $\hat{u}_{|t}$ in accordance with

$$\hat{u}_{|t} = \sum_{i=1}^{t} N_i \nu_i = \hat{u}_{|t-1} + N_t \nu_t \tag{8}$$

It is easily realized that the random sequence $\{e_t\}$ is the innovation process of observations [6,7]. By developing steps i), ii) and iii) the desired recursive estimate is easily obtained [4]

1. $$e_t = z_t - \hat{z}_{t|t-1} \tag{9a}$$

2. $$\nu_t = e_t / \| e_t \| \tag{9b}$$

3. $$N_t = \{M_t - [\langle \hat{u}_{|t-1}, z_k\rangle] / \| e_t \| \tag{9c}$$

4. $$\hat{u}_{|t}=\hat{u}_{|t-1}+N_t\,\nu_t \tag{9d}$$

In eq. (9a) $\hat{z}_{t|t-1}$ represents the best linear prediction of observation z_t on the basis of the past observations up to time (t-1), and

$$\tilde{M}_t \triangleq M_t-[\langle\hat{u}_{|t-1},z_t\rangle] \tag{10}$$

is the one-step prediction error of the cross-correlation vector. It is worthwhile to point out that the components $N_t(i)$ of vector N_t with $i \geqslant L+t$ are all zero, so that at the t-th step, according to eq. (9d), only the first (t+L-1) components of the estimate $\hat{u}_{|t}$ are updated. Moreover, notice that although the components of vectors $\{M_i, i=1,2,\dots\}$ other than zero are only L, this does not happen for vectors $\{N_i, i=1,2,\dots\}$ and so, at each step, all the previous components of the estimate are updated; in other words observations z_t carries with it information about all previous data.

If the measurement model (2) is taken into account the following updating formula can easily be obtained from eqs. (9) [4]:

$$\hat{u}_{|t}=\hat{u}_{|t-1}+K_te_t=\hat{u}_{|t-1}+K_t(z_t-C_t\hat{u}_{|t-1}) \tag{11}$$

with $\hat{u}_{|0}=0$.

By defining the covariance error matrix P_t as

$$P_t \triangleq E[(\hat{u}_{|t}-u)(\hat{u}_{|t}-u)'] \tag{12}$$

the gain vector K_t and matrix P_t can be computed with the recursion

$$K_t=P_{t-1}C_t'[C_tP_{t-1}C_t'+\sigma_n^2(t)]^{-1} \tag{13}$$

$$P_t=P_{t-1}-K_tC_tP_{t-1} \tag{14}$$

with

$$P_o=E(uu') \tag{15}$$

Notice that the term in the square brackets in eq. (13) is a scalar term, and that matrices of dimension (t+L-1)x(t+L-1) are involved in recursion (13,14).

In Fig. 2 we represent a possible way of implementing the estimation algorithm (11). As one can see a continuous updating of the whole estimate is required as time t increases.

APPLICATION TO INTERSYMBOL INTERFERENCE ELIMINATION

The theory developed in the preceding sections may be directly applied to the problem of eliminating the intersymbol interference which arises in data communication through a dispersive channel. In this case vector C_k represents the tap values of the tapped delay line model of the channel and eq. (1) is a representation of the received random sequence. Eq. (11) represents the best linear equalizer while eq. (14) gives the minimum error covariance matrix obtainable. The linear equalizer in Fig. 2 is structured as a Kalman filter equalizer [1] followed by a transversal filter with an infinite number of tap gains, and so it cannot be implemented. In any case the structure in Fig. 2 is the optimal structure for the stochastic linear equalization problem.

More insight in the equalizer structure is obtained if one develops in detail eqs. (13-15). We can easily obtain

$$K_t = a_t^{-1}\left[\sum_{i=2}^{L} c(i)P_{t-1}(1,t+L-i), .., \sum_{i=2}^{L} c(i)P_{t-1}(t+L-2,t+L-i), c(1)\,\sigma_u^2(t), 0, 0, ...\right]' \qquad (16)$$

where the scalar term a_t is defined as

$$a_t \triangleq C_t P_{t-1} C_t' + \sigma_n^2(t) \qquad (17)$$

and $P_t(i,j)$ is the element in the i-th row and j-th column of the matrix P_t. Remember that $P_t(i,j)$ is the cross-covariance between the errors of the estimates at time t of u(i) and u(j) and so, $K_t(i)$, for fixed i, approaches zero as t increases. This means that observations occurring long after a sample has been transmitted receive less weighting in computing the estimate of that sample. Of course, this phenomenon is in accord with what is intuitively reasonable. This discussion leads to a realizable sub-optimal scheme of the linear equalizer as represented in Fig. 3. The truncation step is directly connected with the model of the equivalent discrete channel, and with covariances $\sigma_u^2(k)$ and $\sigma_n^2(k)$. Notice that the structure in Fig. 3 may be regarded as a fixed-lag smoothing equalizer [8].

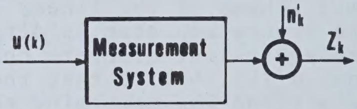
CONCLUSIONS

In this paper we investigated the problem of linear estimating a random digital data sequence from indirect noisy measurements. The problem has been solved by means of Hilbert space methods utilizing the innovations of the observed sequence. Finally, the application of the results to the equalization of data transmissions through dispersive channels was discussed, and we also derived a suboptimal structure of the optimum estimator. Notice that recursion (13-15) does not depend on the received sequence and it may therefore be precomputed and stored. A reduction in storage requirements may be obtained if one utilizes the asympthotic values of gains $K_t(i)$ obtained as $t \rightarrow \infty$. A computer calculation has shown that the asympthotic values are obtained with a number of iterations of the order of 2÷ 3 times L.

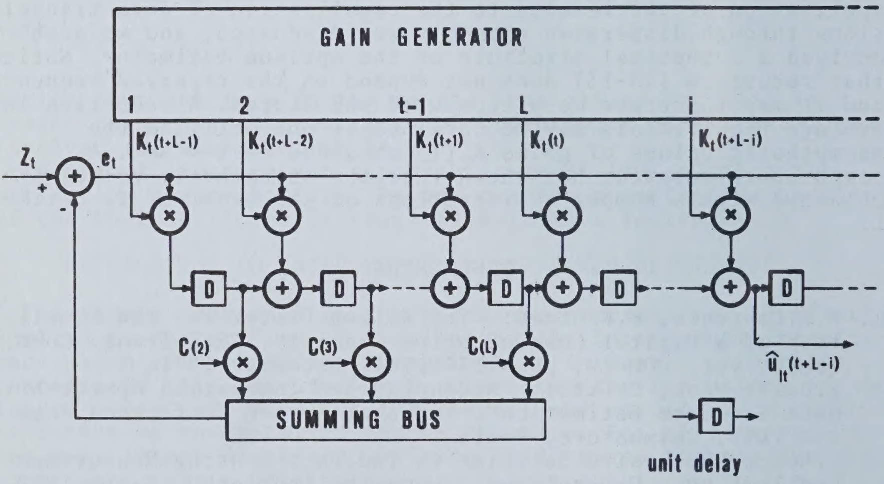
REFERENCES

1. R.E.Lawrence, H.Kaufman: "The Kalman Filter for the Equalization of a Digital Communication Channel", IEEE Trans. Comm. Tech., vol. COM-19, pp. 1137-1141, December 1971
2. E.Dalle Mese, D.Giuli: "A Generalized Innovation Approach to Data Sequence Estimation", National System Conference, June 2-4 1977, Coimbatore, India
3. E.Mosca: "Recursive Solution to Indirect Sensing Measurement Problems by a Generalized Innovation Approach", Sixth IFIP Conf.Optimization Techniques, Novosibirsk, USSR, preprint 11, pp. 1-13, July 1974
4. E.Mosca: "An Innovation Approach to Indirect Sensing Measurements Problems", Ricerche di Automatica, vol. 6, n. 2-3, pp. 1-24, December 1975
5. E.Wong: "Stochastic Processes in Information and Dynamical

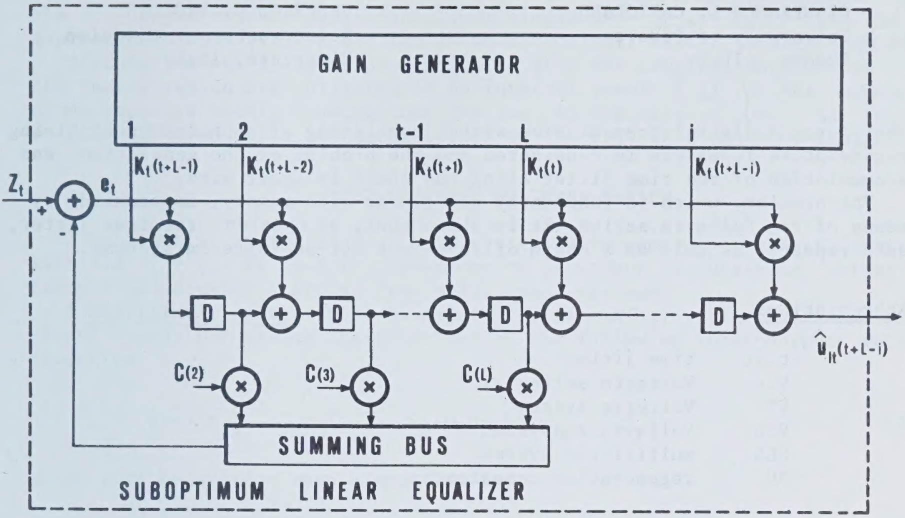
Systems", Mc Graw Hill, New York, 1971
6. T.Kailath: "An Innovations Approach to Least-Squares Estimation. Part I: Linear Filtering in Additive White Noise", IEEE Trans. on Automatic Control, vol. AC-13, pp. 646-655, 1968
7. T.Kailath: "The Innovations Approach to Detection and Estimation Theory", Proc. IEEE, vol. 58, pp. 680-695, May 1970
8. E.Dalle Mese, D.Giuli: "Fixed-Lag Smoother for Digital Channel Equalization", Electronics Letters, vol. 13, n. 12 pp. 366-367, 9th June 1977


u(k)
Measurement System
n'_k
+
Z'_k

- Fig. 1 - Observation model -


GAIN GENERATOR
1
2
t-1
L
i
$K_t(t+L-1)$
$K_t(t+L-2)$
$K_t(t+1)$
$K_t(t)$
$K_t(t+L-i)$
Z_t
e_t
+
-
D
C(2)
C(3)
C(L)
$\hat{u}_{lt}(t+L-i)$
SUMMING BUS
D
unit delay

- Fig. 2 - Optimum minimum error variance linear estimator -


GAIN GENERATOR
1
2
t-1
L
i
$K_t(t+L-1)$
$K_t(t+L-2)$
$K_t(t+1)$
$K_t(t)$
$K_t(t+L-i)$
Z_t
e_t
+
-
D
$C(2)$
$C(3)$
$C(L)$
$\hat{u}_{lt}(t+L-i)$
SUMMING BUS
SUBOPTIMUM LINEAR EQUALIZER

- Fig. 3 - Suboptimum linear equalizer -

A VOLTERRA SERIES APPROACH TO THE TIME JITTER IN A CHAIN OF REGENERATIVE REPEATERS

GIANFRANCO L. CARIOLARO
University of Padova
Padova, Italy

FEDERICO C. VAGLIANI
Telettra Transm. Division
Vimercate, Italy

Abstract: A digital transmission system consisting of a chain of self-timing regenerative repeaters is considered and the problem of the generation and accumulation of the time jitter along the chain is dealt with.

The problem, which is inherently non linear with memory, is tackled by means of the Volterra series. It is shown that, as regards the time jitter, each repeater as well as a chain of repeaters act as Volterra systems.

Abbreviations

t.j.	time jitter
V.s.	Volterra series
VS	Volterra system
VSS	Volterra subsystem
MLS	multilinear system
RR	regenerative repeater

I. INTRODUCTION

Digital data transmission systems over physical lines make use of regenerative repeaters (RRs) which perform the basic functions of reshaping, retiming, and regeneration. These RRs are said to be self-timed since the timing information required in each RR is extracted from the incoming pulse train itself.

The ideal timing information should consist of a sequence of equally-spaced instants, namely $\{nT+t_c\}$, $-\infty < n < +\infty$, where t_c is a fixed time and T is the symbol period. In practice, this sequence is always affected by an error usually called time jitter (t.j.). After the k-th timing extraction in place of the ideal sequence we find:

$$\{nT + t_c + \xi_k(n)\} \quad , \quad -\infty < n < +\infty \quad , \tag{1}$$

where $\xi_k = \xi_k(n)$ represents the t.j. In general, ξ_k is the sum of two contributions: a part that is produced internally during the k-th timing extraction and a part that is caused by the preexisting t.j. ξ_{k-1} in the incoming pulse train. Thus, in a chain of RRs the t.j. is produced in each RR and it accumulates along the chain. The effects of the t.j. may be particularly serious in a long chain because of the possibility of an excess increase of the t.j. along the chain.

Several authors have considered this problem in the past two decades [1 ÷ 8]. However, for obvious reasons of tractability it was often oversimplified; certain effects as the influence of the pulse waveform and the presence of nonlinear processings in the timing circuits, have not received the adequate emphasis. On the other hand, these effects cause often a considerable contribution to the t.j. amount.

The problem is inherently nonlinear with memory and, indeed, in the present paper we show that the Volterra series (V.s.) represents the natural tool to perform a not simplified analysis. In Section II we introduce the simple and the double discrete V.s. (Note that the t.j. $\xi_k = \xi_k(n)$ must be regarded as a discrete-time signal); the simple V.s. serves to describe an one-input one-output nonlinear system with memory, whereas the double V.s. describes a two-input one-output system of the same kind. In Section III we show that the t.j. input-output relationship in a simgle RR, $\xi_{k-1} \rightarrow \xi_k$, can be expressed through V.s. In Section IV we extend this result to a chain of RRs.

Obvious reasons of space prevent us to give the complete derivations of the theory (which are collected in an internal report [11]). The purpose of the paper is merely showing that the V.s. is the natural tool to investigate the production and accumulation of the t.j. in a chain of RRs.

II. THE DISCRETE VOLTERRA SERIES

A substantial number of nonlinear systems with memory can be represented as a V.s. [9]. We find it convenient to introduce the discrete series (simple and double) starting from multilinear systems.

A multilinear system (MLS) of order r, say $\mathscr{M}(h_r)$ (fig. 1), is an r--input one-output system characterized by the following input-output relationship:

$$y(n) = \sum_{m_1} \dots \sum_{m_r} x_1(m_1) \dots x_r(m_r) h_r(n; m_1, \dots, m_r) , \tag{2a}$$

which will be written more compactly as:

$$y = (x_1, \dots, x_r) \circ h_r . \tag{2b}$$

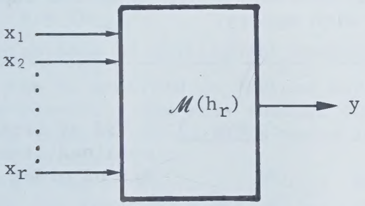


Fig. 1: A multilinear system (MLS) of order r with kernel h_r .

Here, y is the output, $x_1, \dots, x_r$ are the inputs, and the kernel h_r describes the system; $n, m_1, \dots, m_r$ are discrete times which belong to the set of integers. This system is multilinear in the sense that $\mathscr{M}(h_r)$ acts as a linear system with respect to each one of its inputs when the other inputs are fixed.

Two trivial examples of MLSs which will be encountered in the following are depicted in fig. 2. The first one is an r-input multiplier characterized by the following kernel:

$$\begin{aligned} \delta_r(n; m_1, \dots, m_r) &= 1 , \quad n = m_1 = \dots = m_r , \\ &= 0 , \quad \text{otherwise} . \end{aligned} \tag{3}$$

In the second example the r inputs are separately processed by ordinary linear systems $\mathscr{L}(g_k)$ and then multiplied together to form the output signal. The resulting kernel is the product of the impulsive responses of the $\mathscr{L}(g_k)$, i.e.:

$$h_r(n; m_1, \dots, m_r) = g_1(n; m_1) \dots g_r(n; m_r) .$$

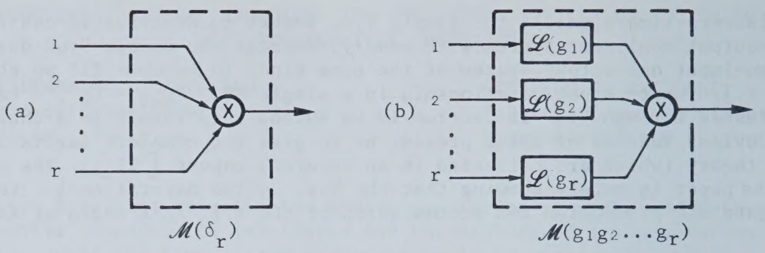


Fig. 2: Examples of MLSs (a) r-input multiplier. (b) cascading of r linear systems $\mathscr{L}(g_k)$ with a multiplier .

A. The simple Volterra series

A Volterra subsystem (VSS) of order r, say $\nu(h_r)$, is obtained by connecting the r inputs of a given MLS $\mathscr{M}(h_r)$ to form a single input (fig. 3). The input-output relationship for $\nu(h_r)$ is therefore:

$$u_r = \underbrace{(a,\dots,a)}_{r\text{ times}} \circ h_r \triangleq (a)_r \circ h_r \quad , \tag{4}$$

where u_r is the output and a the input.

Even if $\mathscr{M}(h_r)$ and $\nu(h_r)$ are characterized by the same kernel h_r, the difference between them is substantial. Indeed, $\nu(h_r)$ is an one-input one-output which in general is nonlinear with memory.

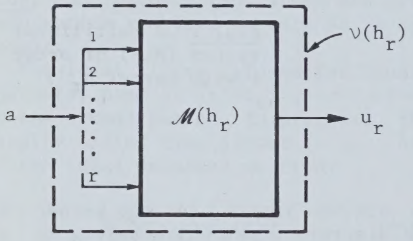


Fig. 3: A VSS of order r obtained from a MLS of order r .

A Volterra system (VS) $\mathscr{V}(h)$ may be viewed as the parallel connection of VSSs $\nu(h_r)$ of any order r . The input-output relationship of $\mathscr{V}(h)$ is therefore:

$$u = \sum_{r=1}^{\infty} (a)_r \circ h_r = V(a;h) \quad , \tag{5}$$

where $V(a;h)$ stands for a V.s. with input a and kernels $h = (h_1, h_2, \dots)$.

It is to be noted that the first-order kernel in (5) characterizes the linear part of the VS, whereas the higher order kernels series describe the various order of nonlinearity.

B. The double Volterra series

Let $\mathscr{M}(h_{rs})$ be a multilinear system of order $r+s$ whose input terminations are grouped into two parts of r and s terminations, respectively. Then, a double VSS, of order r, s say $\nu_2(h_{rs})$, is obtained by connecting together the terminations in each one of the two groups to form two separed input terminations (fig. 4). Hence, $\nu_2(h_{rs})$ is a two-input one output system

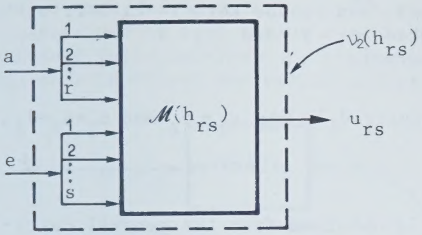


Fig. 4: A double VSS of order (r,s) obtained from a MLS of order r + s.

characterized by the following relationship:

$$u_{rs} = ((a)_r, (e)_s) \circ h_{rs} \tag{6}$$

where u_{rs} is the output, and a and e are the two inputs.

A double VS, $\mathscr{V}_2(h)$, may be viewed as the parallel connection of double VSSs, $\nu_2(h_{rs})$, of any order (r,s). The input-output relationship of $\mathscr{V}_2(h)$ is therefore:

$$u = \sum_{r=1}^{\infty} \sum_{s=1}^{\infty} ((a)_r, (e)_s) \circ h_{rs} = V_2(a,e;h) \tag{7}$$

where $V_2(a,e;h)$ stands for the double V.s. with inputs a and e, and kernels $h = (h_{rs})$.

C. Combinations of multilinear systems

MLSs can be combined in various ways to form new MLSs; the last in turn serve to construct new VSSs through the procedure of grouping input terminals (see figs. 3 and 4). This operation would not be possible in general working directly on VSSs.

Cascading. Let $\mathcal{M}(h_{p1}),\ldots,\mathcal{M}(h_{p_m})$ be m MLSs whose m outputs are connected to the input of a MLS $\mathcal{M}(\lambda_m)$ of order m. The resulting structure (fig. 5) is a new MLS, say $\mathcal{M}(h_{p1}\ldots h_{p_m} \square \lambda_m)$, of order $p_1 + p_2 + \ldots + p_m$, which will be termed the cascading of $\mathcal{M}(h_{p1}),\ldots,\mathcal{M}(h_{p_m})$ with $\mathcal{M}(\lambda_m)$. The new kernel is easily expressed in terms of the original kernels. For instance, in the case of fig. 5 where $p_1 = 2$, $p_2 = 3$, and $m = 2$, we get:

$$\begin{aligned} h_2 h_3 \square \lambda_2(n;c_1,c_2,d_1,d_2,d_3) &= \\ = \sum_{n_1} \sum_{n_2} h_2(n_1;c_1,c_2) h_3(n_2;d_1,d_2,d_3) \lambda_2(n;n_1,n_2) \end{aligned} \tag{8}$$

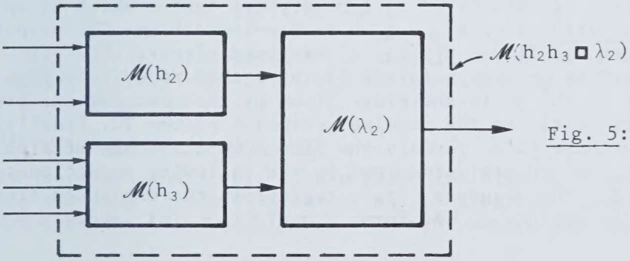


Fig. 5: Cascading of $\mathcal{M}(h_2)$, $\mathcal{M}(h_3)$ with $\mathcal{M}(\lambda_2)$.

Pre-product of $\mathscr{M}(\lambda_m)$. It is obtained as a particularization of the previous cascading when the m MLSs are replaced by m multipliers of the corresponding orders, i.e. when $h_{p1} = \delta_{p1},\dots,h_{pm} = \delta_{pm}$. The resulting kernel is then specified by the m-tuple $\underline{p} = (p_1,\dots,p_m)$ and by the kernel λ_m . It will be denoted by $\mathscr{D}(\underline{p})\lambda_m$. For instance, when $p_1 = 2$ and $p_2 = 3$ the pre-product of $\mathscr{M}(\lambda_2)$ has the fifth order kernel:

$$\begin{aligned}\mathscr{D}(2,3)\lambda_2(n;c_1,c_2,d_1,d_2,d_3) &= \lambda_2(n;c_1,d_1), \quad \text{for } c_1 = c_2 \text{ and } d_1=d_2=d_3 ,\\ &= 0 \qquad\qquad\qquad\;\; , \text{ otherwise} \quad .\end{aligned}$$

Post-product of $\mathscr{M}(h_{p1}),\dots,\mathscr{M}(h_{pm})$. It is obtained from the general cascading structure when $\mathscr{M}(\lambda_m)$ is a multiplier. The resulting kernel will be simply denoted by $h_{p1}h_{p2}\dots h_{pm}$. In the reference case ($p_1=2$, $p_2=3$) we get explicitely:

$$h_2h_3(n;c_1,c_2,d_1,d_2,d_3) = h_2(n;c_1,c_2)h_3(n;d_1,d_2,d_3) \quad .$$

III. TIME JITTER RELATIONSHIP IN A REGENERATIVE REPEATER

In this section we establish the input-output t.j. relationship, $\xi_{k-1} \to \xi_k$, in the k-th RR of a chain. We set for brevity:

$$e(n) \triangleq \xi_{k-1}(n) \text{ : input t.j. ,} \quad u(n) \triangleq \xi_k(n) \text{ : output t.j.}$$

A. Timing extraction circuit

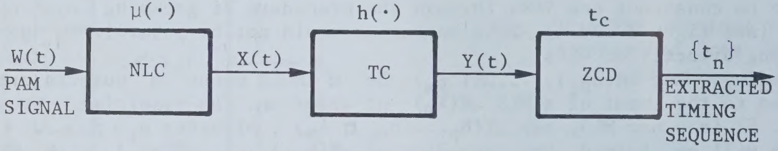


Fig. 6: Timing extraction circuit.

We assume that the timing extraction is performed through the system of fig. 6. The driving signal, which is taken from the decision point of the RR, has the PAM jittered representation:

$$W(t) = \sum_n a(n)g(t - nT - e(n)) \tag{9}$$

where $a = a(n)$ is the digital data sequence and $g(t)$ is a standard pulse waveform. The nonlinear circuit (NLC) is a memoryless system which is specified by a characteristic $X = \mu(W)$, e.g. a square-law device. The output of NLC is therefore given by $X(t) = \mu[W(t)]$. The tuned circuit (TC) is a narrowband filter tuned as close as possible to the symbol repetition rate $f_c \triangleq 1/T$. The output of the TC is therefore given by the convolution: $Y(t) = X(t) * h(t)$, where $h(t)$ is the impulse response of the TC. Finally, the zero-crossing detector (ZCD) reveals the zero down-crossings of $Y(t)$, i.e. the instants t_n which are determined by the following conditions: $Y(t_n) = 0$, $\dot{Y}(t_n) < 0$. The sequence $\{t_n\}$ represents the extracted timing information and can be written in the form: $t_n = nT + t_c + u(n)$ where $u = u(n)$

is the output t.j.

Note that ideally the timing wave $Y(t)$ should be a perfect sinusoid with frequency $f_c \triangleq 1/T$. In practice, $Y(t)$ exhibits both a random phase and an amplitude modulation which cause the presence of the output t.j. $u(n)$.

If we assume that timing circuit parameters μ , h, t_c, as well as the standard pulse waveform g , are assigned, the system of fig. 6 may be regarded as a two-input one-output system. The inputs are the data sequence a

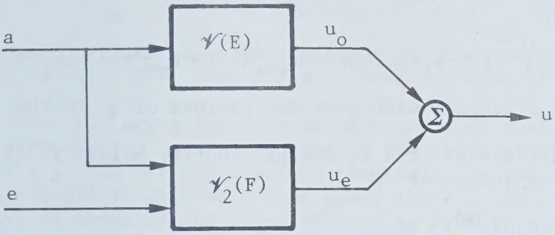


Fig. 7: Volterra system representation of a RR with respect to the t.j.

and the input t.j. e, whereas the output is the output t.j. u . The relationship $(a,e) \rightarrow u$ can be expressed through the V.s. as follows:

$$u = u_o + u_e = V(a;E) + V_2(a,e;F) \quad . \tag{10}$$

Thus, as regards the t.j. a RR may be viewed as a parallel connection of a single VS and a double VS (fig. 7). Note that in the absence of input t.j. $(e = 0)$, we find from (10) $u_e = 0$, and $u = u_o$. Thus, u_o is the amount of t.j. produced internally in the RR and u_e is the amount caused by the input t.j. e .

We now mention how (10) can be derived and we relate the kernels E_r and F_{rs} to the timing circuit parameters.

B. Output jitter in terms of the timing wave

After a suitable choice of the reference instants $nT + t_c$ [6], the output t.j. turns out to be uniquely determined by the condition: $Y(t_n) = Y(nT + t_c + u(n)) = 0$.

Now, the first-order approximation to $u(n)$, say $u_1(n)$, is obtained by expanding $Y(t)$ around $t = nT + t_c$ and neglecting higher order terms. We find $Y(nT + t_c) + \dot{Y}(nT + t_c)u_1(n) = 0$. Thence $u_1 = -y/\dot{y}$, where $y = y(n) \triangleq Y(nT + t_c)$ and $\dot{y} = \dot{y}(n) \triangleq \dot{Y}(nT + t_c)$.

Hence, working on the inverse of the function $Y(t)$ in the neighborhood of $t = nT + t_c$ it is possible to state a series expansion of the output t.j. u in powers of the first-order approximation u_1 .

Theorem 1 [11]: If the timing wave $Y(t)$ has derivatives of any order m at $t = nT + t_c$, say $y^{(m)} = y^{(m)}(n) \triangleq Y^{(m)}(nT + t_c)$, the output t.j. is given by $u = u_1 + u_2 + u_3 + \dots$, where

$$u_k = Z_{k-1}(z_2,\dots,z_k)u_1^k/k! \ , \quad z_m \triangleq -y^{(m)}/\dot{y} \quad . \tag{11}$$

The $Z_{k-1}(\cdot)$ are multivariable polynomials which are related to the Bell polynomials [10]. The first few instances of $Z_{k-1}(\cdot)$ are: $Z_o = 1$, $Z_1(z_2) = z_2$, $Z_2(z_2,z_3) = z_3 + 3\,z_2^2$.

C. Expansion of the timing wave

In Theorem 1 the t.j. is expressed in terms of the sample values of the timing wave and its derivatives, i.e. $y \triangleq Y(nT + t_c)$ and $y^{(m)} \triangleq Y^{(m)}(nT + t_c)$. We now state the expansion for y and $y^{(m)}$, which are obtained working with di Bruno's formula [9] to express the derivatives of composite functions, as $X(t) = u[W(t)]$. The assumptions are that the functions g, h, and μ (see fig. 6) have derivatives of any order.

Theorem 2 [11]: The sequence of the timing wave samples $y(n) = Y(nT + t_c)$ has the following expansion

$$y = y_o + y_e = V(a;A) + V_2(a,e;B) \quad . \tag{12}$$

The kernels A are defined by:

$$A_r(n;m_1,\ldots,m_r) \triangleq \frac{\mu^{(r)}(0)}{r!} \int_{-\infty}^{+\infty} g(\tau - m_1 T)\ldots g(\tau - m_r T)h(nT + t_c - \tau)d\tau \quad . \tag{13}$$

The kernels B have a similar structure with some derivatives of g in the integrand.

Corollary 1 : From $Y(t) = X(t) * h(t)$ the following identity holds: $Y^{(m)}(t) = X(t) * h^{(m)}(t)$. Then, the expansion of $y^{(m)}$ is:

$$y^{(m)} = y_o^{(m)} + y_e^{(m)} = V(a;A^{(m)}) + V_2(a,e;B^{(m)}) \quad , \tag{14}$$

where $A^{(m)}$ and $B^{(m)}$ are obtained from A and B by replacing h with $h^{(m)}$.

D. Expansion for the output time jitter

By combination of the previous results it is possible to derive the expansion of the t.j. as stated in (10). The main difficulty arises from the fact that in t.j. series a preliminar expansion of the ratios $u_1 = -y/\dot{y}$ and $z_m = -y^{(m)}/\dot{y}$ is to be found. At the time being we have solved this problem by decomposing the slope $\dot{y} = \dot{y}_o + \dot{y}_e$ as $\dot{y} = \dot{m}_o + \Delta\dot{y}_o + \dot{y}_e$, where $\dot{m}_o \triangleq E[\dot{y}_o]$ is the statistical average of $\dot{y}_o$, $\Delta\dot{y}_o$ is the slope fluctuation in the absence of input t.j., and $\dot{y}_e$ is the slope fluctuation caused by the input t.j. This expansion, however, is very complicated since it involves a triple V.s. and its presentation goes beyond the scope of the paper. We limit ourselves giving an approximated solution which follows when the flutuactions in $\dot{y}$ are neglected. The accuracy in this approximation is in part discussed in [6]; on the other hand, recent calculations of the variance of the time jitter in some typical situations have shown that the error in this approximation is confined to within a few percent.

Under the approximation $\dot{y} \simeq \dot{m}_o \triangleq -1/K$,the desired expansions are directly given by:

$$u_1 = V(a;C) + V_2(a,e;D) \quad , \quad z_m = V(a,C^{(m)}) + V_2(a,e;D^{(m)}) \quad , \tag{15}$$

where $C_r = K A_r$, $D_{rk} = K B_{rk}$, $C_r^{(m)} = K A_r^{(m)}$, $D_{rk}^{(m)} = K B_{rk}^{(m)}$.

Hence, after use of (15) in Theorem 1 the kernels in (10) can be found by inspection. The first few instances are:

$$\begin{aligned} &E_1 = C_1, \; E_2 = C_2, \; E_3 = \tfrac{1}{2} C_1 C_1 C_1^{(2)} \;, \\ &F_{11} = D_{11}, \; F_{21} = D_{21}, \; F_{31} = D_{31} + C_1^{(2)} C_1 D_{11} + \tfrac{1}{2} D_{11} C_1^{(2)} C_1 \;, \\ &F_{12} = D_{12}, \; F_{22} = D_{22}, \; F_{32} = D_{32} + \tfrac{1}{2} C_1^{(2)} D_{11} D_{11} + D_{11}^{(2)} C_1 D_{11} \;, \\ &F_{13} = D_{13}, \; F_{23} = D_{23}, \; F_{33} = D_{33} + \tfrac{1}{2} D_{11}^{(2)} D_{11} D_{11} \;. \end{aligned} \tag{16}$$

IV. TIME JITTER RELATION IN A CHAIN OF REGENERATIVE REPEATERS

The input-output relationship established in Section III for a single RR can be used to find the input-output relationship for a chain of an arbitrary finite number of RRs. As usual, we assume that the data sequence $a = a(\overline{n})$ remains unchanged along the chain. Then, the input-output relationship for the m-th regenerator is

$$\xi_m = V(a;E^{(m)}) + V_2(a,\xi_{m-1};F^{(m)}) \quad , \qquad (17)$$

where ξ_{m-1} and ξ_m are the input and the output t.j., respectively, and the kernels $E^{(m)}$ and $F^{(m)}$ are characteristic of the timing circuit of the m-th RR .

Now, for any given data message a , (17) enables us to calculate recursively the t.j. at any section of the chain starting from the knowledge of the t.j. ξ_o at the input of the chain. On the other hand, we can establish an overall relationship that gives the t.j. at the end of the chain, say ξ_M , in terms of the input t.j. ξ_o . This relationship exhibits the same structure as in a single RR, specifically:

$$\xi_M = V(a;E^{(1...M)}) + V_2(a,\xi_o;F^{(1...M)}) \qquad (18)$$

where the overall kernels $E^{(1...M)}$ and $F^{(1...M)}$ depend on $E^{(m)}$ and $F^{(m)}$, $1 \leq m \leq M$ (see below).

Usually, at the input of the chain we find an exact timing information, i.e. $\xi_o = 0$. In this case (18) yields:

$$\xi_{Mo} = V(a;E^{(1...M)}) \qquad (19)$$

which represents the t.j. produced and accumulated along the chain.

A. Structure of the overall kernels

We now prove overall relation (18) and we determine the corresponding kernels. There is no loss of generality if we assume $M = 2$; in other words we limit ourselves showing that a cascading of two RRs (fig. 8) has the

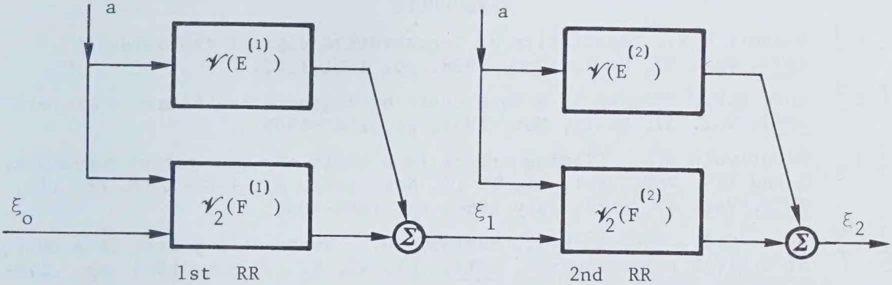


Fig. 8: Volterra system representation for the t.j. in a chain of two RRs .

same equation as a single RR.

Using (17) for $m = 1$ and $m = 2$ and then combining we find by inspection:

$$\xi_2 = V(a;E^{(12)}) + V(a,\xi_o;F^{(12)}) \qquad (20)$$

where the new kernels may be computed through the cascading operator defined in Section III. We get:

$$E_i^{(12)} = \sum_{p_1+\dots+p_m+r=i} E_{p_1}^{(1)}\dots E_{p_m}^{(1)} \,\square\, F_{rm}^{(2)} , \qquad (21a)$$

$$F_{ij}^{(12)} = \sum_{p_1+\dots+p_m+r=i,\ q_1+\dots+q_m=j} F_{p_1q_1}^{(1)}\dots F_{p_mq_m}^{(1)} \,\square\, F_{rm}^{(2)} . \qquad (21b)$$

Note that in both (21) the cascading operator $\square$ acts only on the second set of m inputs of the MLS $\mathscr{M}(F_{rm}^{(2)})$. When $\xi_o = 0$ the computation of the kernels $F^{(12)}$ is not required since in this situation the cascading of two RRs is completely characterized by $E^{(12)}$ (see (19)).

V. CONCLUSION

A systematic approach to the analysis of the t.j. in a single RR and in a chain of RRs has been carried out by means of the V.s. No simplification has been introduced in this analysis (the approximation in the slope $\dot{y}$ was motivated for the lack of space). Indeed, the timing circuit parameters g, μ , and h were assumed to be completely general. Moreover, the downcrossing position was determined as an infinite series which yields the exact position in the limit. Of course, the resulting theory appears to be a formidable one and suitable simplifications must be introduced to get explicit results of practical interest. The important fact, however, is that reasonable simplifications are only possible a posteriori, i.e. when a general exact theory has been developed.

We finally note that the problem of t.j. is long to be solved after the present theory. Indeed, the ultimate interest lies on a statistical evaluation of the t.j. itself, where the t.j. sequence as well as the data sequence are regarded as discrete parameter stochastic processes. Thus, the fundamental problem to be solved is the statistical analysis at the output of a VS. We are now investigating in this area where we have found some preliminary results [12] .

REFERENCES

[1] Bennett W.R., "Statistics of regenerative digital transmission", BSTJ, Vol. 37, N. 11, Nov. 1958, pp. 1501-1542.

[2] Rowe H.E., "Timing in a long chain of regenerative binary repeaters", BSTJ, Vol. 37, N. 11, Nov. 1958, pp. 1543-1598.

[3] Kinariwala B.K., "Timing errors in a chain of regenerative repeaters, I and II", BSTJ, Vol. 41, N. 11, Nov. 1962, pp. 1769-1797. Pt. III, BSTJ, Vol. 43, N. 7, July 1964, pp. 1481-1504.

[4] Byrne C.J., Karafin B.J., Robinson D., "Systematic jitter in a chain of digital regenerators", BSTJ, Vol. 42, N. 11, Nov. 1963, pp. 2679-2714.

[5] Manley J.M., "The generation and accumulation of timing noise in PCM systems. An experimental and theoretical study", BSTJ, Vol. 48, N. 3, March 1969, pp. 541-613.

[6] Bellato L., Cariolaro G.L., "Time jitter in line regenerators with pattern dependent pulse waveforms", Alta Frequenza, Vol. 41, N. 11, Nov. 1972, pp. 800-815.

[7] Franks L.E., Bubrowski J.P., "Statistical properties of timing jitter in PAM timing recovery scheme", IEEE Transactions on Communications, Vol. 22, N. 7, July 1974, pp. 913-920.

[8] Cariolaro G.L., Todero F., "A general spectral analysis of time jitter produced in a regenerative repeater", IEEE Trans. on Communications, Vol. COM. 25, April 1977, pp. 417-426.

[9] Bedrosian E., Rice S.O., "The output of Volterra systems (Nonlinear systems with memory) driven by harmonic and Gaussian inputs", Proc. IEEE, Vol. 59, Dec. 1971, pp. 1688-1707.

[10] Riordan J., "Combinatorial Identities", New York, John Wiley, 1967, p. 181.

[11] Cariolaro G.L., Vagliani F.C., "A Volterra series approach to the time jitter in a chain of regenerative repeaters", CESP Internal Report, October 1977.

[12] Cariolaro G.L., Di Masi G., "Statistical Analysis after a Volterra series transformation driven by a zero-memory source", to be published.

NOISELESS OVERFULL TREE-ENCODING OF MEMORYLESS SKEWED BINARY SOURCES

LEONARDUS TH. N. M. MULDER and DAVID L. COHN
University of Notre Dame, Dept. Electr. Engrg., Notre Dame IN 46556

Abstract

We present here some preliminary results obtained for a generalized runlength coding scheme for skewed binary sources. We call these codes runlength overfull tree codes, having n' runlength messages and n codewords, where $n' \geq n$. *

Introduction

An Optimal variable-to-fixed noiseless source-coding algorithm has been developed earlier by Tunstall[1] in the case that the number of codewords is equal to the number of messages. Later, Cohn[2] showed how overfull tree codes, having more messages than code words in some cases may result in a lower rate. This paper is an extension of Cohn's research for the case of skewed binary sources. Methods for optimizing overfull tree codes with a run-length message-set are described for a fixed number of codewords n.

1-Full and Overfull Tree codes

In varaible-to-fixed noiseless source coding we can represent the set of encoded source sequences by a c-ary tree, the so-called enocdingtree. These encoded source sequences are called messages and form the messageset. Here c denotes the alphabetsize of the source, so c=2 is the case we consider here. Related to the encoding tree is the concept of instantaneous encodability. This simply means that as soon as a source sequence is recognized by the encoder as a message it will be encoded. Codes for which an encodingtree and thus a messageset is defined are therefore always instantaneous encodable. Similarly we can define the decodingtree, whose terminal nodes represent codewords. A codeword is the smallest channel sequence that is accepted by the decoder and is decoded into a source sequence, its codeword-image. Related to the decodingtree is the concept of instantaneous decodability, often called block-decodability. Completely analogous to instantaneous encodability, block-decodability means that as soon as a channel sequence is recognized by the decoder as a code word it will be decoded.

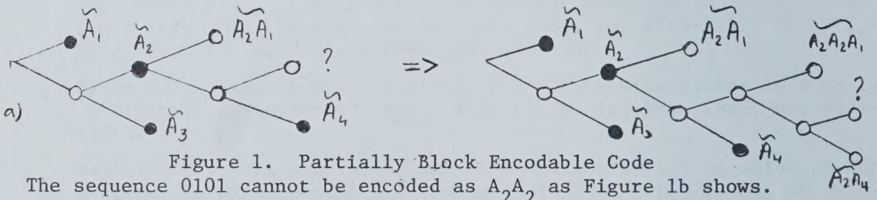


Figure 1. Partially Block Encodable Code

The sequence 0101 cannot be encoded as A_2A_2 as Figure 1b shows.

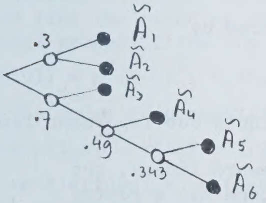
The sequence 010100 will be encoded as A_2A_4 not having A_2A_2 as a prefix. n = 4 but n' is not defined.

---

*This work was supported by National Science Foundation, Grant No. NSF ENG 75/21031

We will not completely derive all the formal concepts related to noiseless source coding as is done in detail in Cohn[2]. But we will briefly mention the concept of partial block encodability, pertaining to source sequences that can only be aprtially encoded. Codes consisting of partially encodable but not block or instantaneous encodable source sequences do not have a messageset but may be encodable with finite delay as an example in figure 1 shows.

In this paper we will restrict ourselves to those codes that are both block encodable and block decodable. These codes are called block-to-block codes. For these codes the messages are encoded as strings of codewords. We will denote the number of messages by n' and the number of codewords by n. Also we call the codewordstring in which a message is encoded the message-image. So we also have n' message-images and n codeword-images. Cohn[2] showed that we only need to consider $n' \geq n$. The case $n' = n$ is well known and leads to the definition of full-tree codes. An optimal variable-to-fixed algorithm for full-tree codes has been found earlier by Tunstall[1]. This algorithm forms the encoding tree by succesively extending the most probable node in the endoding tree untill the number of terminal nodes equals n. An example that illustrates the procedure is given in figure 2.



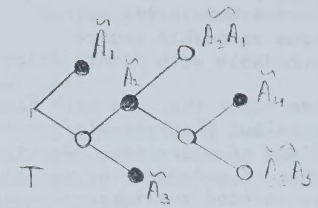
Messages	Code Words
11	A_1
10	A_2
01	A_3
001	A_4
0001	A_5
0000	A_6

Figure 2. Illustration of Tunstall's algorithm for $p = .3$, $n' = 6$.

Note that {codewords} ≡ {message-images}
while {messages} ≡ {codeword-images}
for full-tree codes.

An optimal variable-to-variable algorithm for full-tree codes has not yet been found, though a suboptimal algorithm is presented in Mulder and Cohn[5]. The decoding-tree in this case is the well-known Huffman tree, based on Huffman's algorithm of succesively combining the nodes with lowest probability in the decoding tree. In variable-to-fixed algorithms we assign a fixed length n_F, given by equation (3) to each codeword. We will deal in this paper with the case[*] $n' \geq n$, leading to the concept of overfull-tree codes, a name originally given to these codes by Cohn[2], referring to the fact that these codes have more messages than codewords. Figure 3 shows an example in which $n' = 5$ and $n = 4$. We restrict ourselves to variable-to-fixed overfull-tree codes for skewed binary memoryless sources. These sources are characterized by a source probability p, being the probability of a source-output 1.

*) $n'=n$ overfull-tree code exists, viz the shortest 2-letter tree (see figure 6), but can always be improved by the full tree.



Messages	Message Image	Code Word	Codeword Image
1	A_1	A_1	1
011	A_2A_1	A_2	01
0101	A_4	A_3	00
0100	A_2A_3	A_4	0101
00	A_3		

Figure 3. Block Encodable Code $n = 4$, $n' = 5$

T is an overfull tree.

For these sources the entropy-function h(p) is given by

$$h(p) = -p\log_2 p - (1-p)\log_2 (1-p) \tag{1}$$

For overfull-tree codes we prove the following simple but important lemma.

Lemma 1 If all codewords have non-null codeword images, the encoding tree of an overfull-tree code has a rooted full subtree whose terminal nodes correspond to codeword-images only.

Proof: Start at the rootnode of the encoding tree and test the intermediate nodes. If an intermediate node corresponds with a codeword-image then remove that part of the encoding tree that extends beyond this node. Continuing this way results in either one of the following two cases, *viz.*
i) no terminal nodes remain, which means that the tree had a rooted full subtree whose terminal nodes were all codeword-images, completing the proof, or
ii) some terminal nodes remain, representing messages that apparently do not have a codeword-image as a proper prefix but then the message itself has to be a codeword-image. If null codeword-images were allowed, these terminal nodes could be represented by strings of codewords. QED

From now on we will denote the encoding tree by T and its rooted full subtree by T'. Note that T' corresponds with a full-tree code. The rooted full subtree T' as obtained in the proof is unique. However, this does not mean that no other rooted full subtree with codeword images as terminal nodes exist as an example in figure 4 shows.

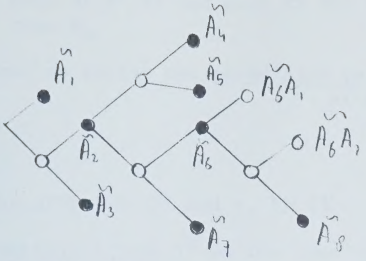


Figure 4. Encoding tree with two rooted full subtrees, vi2 T' $(\tilde{A}_1,\tilde{A}_2,\tilde{A}_3)$ and $T^*(\tilde{A}_1,\tilde{A}_4,\tilde{A}_5,\tilde{A}_6,\tilde{A}_7,\tilde{A}_3)$.

Since the messages for a memoryless source are independent, the message probabilities can be easily calculated. The codeword-probabilities can be found from the encoding tree. As an example, consider figure 3. The message-probabilities are respectively $P_r(1) = p$, $P_r(011) = p^2(1-p)$, $P_r(0101) = p^2(1-p)^2$, $P_r(0100) = p(1-p)^3$ and $P_r(00) = (1-p)^2$. We then find for the codeword probabilities, $P(A_1) = p + 1/2\, p^2(1-p)$, $P(A_2) = 1/2\, p^2(1-p) + 1/2\, p(1-p)^3$, $P(A_3) = 1/2\, p(1-p)^3 + (1-p)^2$ and $P(A_n) = p^2(1-p)^2$. From now on, to distinguish between codeword-probability $P(A_i)$ and the probability of the codeword-image of A_i, we will denote the latter by $P(\tilde{A}_i)$. Also $P(\tilde{A}_i\tilde{A}_j)$ denotes the probability of the node having assigned the string of codewords A_iA_j and the node itself will be indicated as $\tilde{A}_i\tilde{A}_j$. As shown in reference 2, the average number of channel-digits required per source-output called the rate of the code and denoted by R, is given by equation (2).

$$R = \frac{\sum_{i=1}^{n'} P_i n_i}{\sum_{i=1}^{n'} P_i \ell_i} \tag{2}$$

Where: P_i = probability of ith message
n_i = number of channel digits in ith message-image
ℓ_i = number of source digits in ith message

For variable-to-fixed coding each codeword has a fixed number of channel digits n_F, given by

$$n_F = \lceil (\log_D n) \rceil \tag{3}$$

Where D is the alphabetsize of the channel. We will again consider the binary case D=2.

Consequently, according to the definition of block-to-block codes we can write for the number of channel digits in the ith-message

$$n_i = (1 + m_i)\, n_F \tag{4}$$

where $m_i \geq o$, m_i integer.

Substituting (4) in (2) results in

$$R = \frac{(1 + r).\, n_F}{L} \tag{5}$$

where

$$r \triangleq \sum_{i=1}^{n'} P_i\, m_i \tag{5a}$$

$$L \triangleq \sum_{i=1}^{n'} P_i\, \ell_i \tag{5b}$$

Jelinek and Schneider[4] have shown that L can be determined by summing up the probabilities of all intermediate nodes of the encodingtree, including the root-node with probability 1. For full-tree codes m_i=o for all i and hence r=o. Therefore maximizing L by extending succesively the most probable node in the encodingtree yields the optimal full-tree code, viZ the code with lower R. This is exactly Tunstall's algorithm.

With an over full-tree code it is possible to increase L beyond the value obtained by the optimal full-tree code for the same number of codewords n. However, this is at the expense of an increase in r from zero to some positive value. Whether the overfull-tree code will be better than the full-tree code therefore depends on the trade-off between r and L.

In the next section we discuss runlength-type over full-tree codes for the skewed ($p \ll o.5$) memoryless binary source.

2. Runlength binary overfull tree-codes

The overfull-tree codes discussed in this section will be said to be of runlength type, because all messages are of the form $o^i 1$,ie. i zero's followed by a one. Bradley[3] appears to be the first to have used such codes and we will show that his code has an optimal structure, but still can be improved by further extension. Let n-k be the number of terminal nodes in the full subtree T'. We will call the remaining k nodes, corresponding to the k remaining codeword-images copious nodes, referring to the fact that they represent an abundant number of zero's. Let the full subtree T' be fixed as a runlength full tree. Then the codeword-images of T' are 1,o1,oo1,....., $o^{n-k-2}1$ having associated codewords A_1, A_2,..., A_{n-k-1} and o^{n-k-1} having associated codeword A_{n-k}. In what follows we assume that the source-probability p is small enough to garantee that the node corresponding to A_n is the most probable node. The next lemma shows that the runlength structure of the overfull-tree code results in a minimal rate R. First we define the average length L_o of T', similarly as in (5b), by (6a) and its rate R_o by (6b).

$$L_o \triangleq \sum_{i=1}^{n-k} P_i\, \ell_i \tag{6a}$$

$$\hat{R}_o \triangleq 1/L_o \tag{6b}$$

Lemma 2 An optimal overfull tree T with a full runlength subtree T' extends beyond the most probable terminal node of T' if and only if the rate of T is lower than R_o.

To shorten notation define the normalized rate $\hat{R}$ by

$$\hat{R} = R/n_F \tag{7}$$

Where R is given by (5) and n_F by (3).

We define an extension tree T" as the tree that extends beyond the full subtree T'. In other words, T" contains the k copious nodes, while T' contains (as its terminal nodes) the n-k non-copious nodes. We have assumed here that the overfull tree T only extends beyond one node in T'. The proof of lemma 2 is easily modified in T'. Suppose we assign temporarily the codeword Y to the root of T" and consider it as a rooted tree that is, having root probability 1. Since T" does not recessarily have a full rooted subtree whose terminal nodes are codeword-images we cannot directly define its rate. But from information of T' we can again 'label' the terminal nodes as strings of codewords of length m_i+1, so we have

$$r_e \overset{\Delta}{=} \sum_{i=1}^{n''} P_i\, m_i \qquad \text{where } n'' = \text{number of terminal rodes in T}'' = n' - (n-k-1)$$

Also we have

$$L_e \overset{\Delta}{=} \sum_{i=1}^{n''} P''_i\, \ell''_i$$

where P''_i = probability of ith terminal node in T".
ℓ''_i = length of path from Y to ith terminal node.

Again we can find L_e by summing up the probabilities of the intermediate nodes of T", including the root-node probability 1. We now continue with the proof of lemma 2.

Proof lemma 2 Consider a particular tree T obtained by connecting T" at a terminal node $\tilde{A}_i$ of T'. As has been shown in reference 5, we find for the average length L of T, $L=L_o+ P(\tilde{A}_i).L_e$

In the same way (replace $\tilde{Y}$, the root of T" by $\tilde{A}_i$, 'scaling' the terminal node-probabilities by a factor $P(\tilde{A}_i)$), we find for r, defined by (5a)

$$r= P(\tilde{A}_i).r_e$$

Therefore the normalized rate of T is given by

$$R_T=(1+P(\tilde{A}_i).r_e)/(L_o+P(A_i).L_e)$$

Requiring $\hat{R}_T<1/L_o$ results in $r_e<L_e/L_o$

Now taking the derivative of R_T with respect to $P(\tilde{A}_i)$ yields

$$\frac{dR_T}{d(P\tilde{A}_i)} = \frac{r_e L_o - L_e}{L^2} < o$$

Therefore we should extend from the most probable node, completing the proof. Figure 5 gives an illustration. QED

We will call an encodingtree whose message-images are strings at most in codewords an m-letter tree. So T' is a 1-letter tree, as is any full encodingtree. The shortest 2-letter tree has n'=n messages as an example in figure 6 shows, though it is clearly inferior to the full 1-letter tree with n messages.

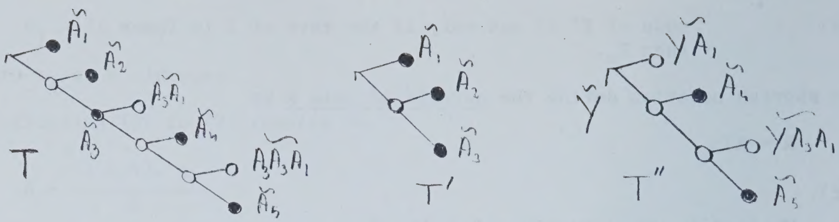


Figure 5. Illustration of decomposition of encoding tree T in full subtree T´ and extension tree T".

The shortest 2-lettertree can be successively extended until we reach the longest 2-letter tree as stated in lemma 3.

Lemma 3 The optimal 2-letter runlength tree is the longest 2-letter tree with (k+1)(n-k+1)+1 messages.

Proof: Consider a 2-letter tree with n+j messages ($o \leq j < k(n-k)+2$) Let $R_1=(1+r_1)/L_1$ and consider one extension from $\tilde{A}_n$, leading to a 2-letter tree with n+j+1 messages and having a normalized rate $\hat{R}_2=(1+r_1+p.P(\tilde{A}_n))/(L_1+P(\tilde{A}_n))$. Then $\hat{R}_2<\hat{R}_1$ iff $\hat{R}_1>p$. We will now show that $\hat{R}_1>p$. For L_1 we find (let $s=n+j$ and $q=1-p$). $L_1=1+q+q^2+\ldots+q^{s-2}=(1-q^{s-1})/P$ While for r_1 we find, noting that all terminal nodes, except for those associated with $A_1, A_2, \ldots, A_{n-k-1}$ and A_n contribute to r_1, $r_1=q^{n-k-1}-q^{s-1}$.

Therefore $\hat{R}_1=p.(1+q^{n-k-1}-q^{s-1})/(1-q^{s-1})>p$ QED

Unfortunately, the result in lemma 3 does not generalize for m>2. However, in the case k=1 it turns out that the optimal over full encoding tree is a 'multiple concatenation' of its full subtree, as stated in lemma 4.

Lemma 4 The optimal runlength overfull encoding tree with n codewords and one copious node has m(n-2)+1 messages, where m is the unique positive integer satisfying

$$(1-Z^{m+1})/(1-Z) \leq m < (1-Z^m)/(1-Z)+1 \tag{8}$$

Where $Z=(1-p)^{n-2}$, and has a rate R given by

$$R=[1/(1-Z)-(m-1)/(Z^{-m}-1)]\cdot p\cdot n_F \tag{9}$$

Proof: Consider the m-letter tree T(m-1,j) with $(m-1).(n-2)+j+1$ messages. It follows then that $L=(1-Z^{m-1}.q^j)/p$, while for r we find

$$r=p\sum_{i=1}^{m-2} i.Z^i\sum_{\ell=o}^{n-3} q^\ell + p(m-1)\ Z^{m-1}\sum_{\ell=o}^{j-1} q^\ell$$

and therefore

$$R(m-1,j) = \frac{1+r}{L}.p.n_F= \frac{[(1-Z^m)/(1-Z) - (m-1)\ Z^{m-1}.q^j]}{1 - Z^{m-1}.\ q^j}.p.n_F \tag{10}$$

We now show that R (m-1,j) is minimized for either j=o or j=n-2. Suppose first j<n-2 and consider one tree T(m-1,j+1).

We find $R(m-1,j+1)-R(m-1,j)=(1-Z^m)/(1-Z)].f(p,m,j).n_F$, where f(p,m,j) is positive for all o<p<1,

therefore, if $(1-Z^m)/(1-Z)>(m-1)$ (10a)

Then $R(m-1,j+1)$ is smaller than $R(m-1,j)$ and we can keep on increasing j until $T(m-1,j+1)$ $(=T(m,o))$ is obtained with $m(n-2)+1$ messages and a rate $R(m,o)$ is given by (8). Increasing further leads to an $(m+1)$-letter tree. Now $R(m,j+1) \geq R(m,j)$, iff

$$(1-Z^{m+1}) \;/\; (1-Z) \leq m \tag{10b}$$

Then we can keep on decreasing j until $j=o$. This shows that the optimal runlength overfull encoding tree with one copious node will have $m(n-2)+1$ messages, for some m. Besides, by combining (10a) and (10b) we obtain equation (8). We will now show that for given Z a unique integer m exists that satisfies (8) and therefore optimizes the code.

Let $S_m = \sum_{i=o}^{m} Z^i$

Then $S_1>1$. Since S_m increases by $Z^{m+1}<1$ when m increases by 1, there will be a unique positive integer m_o, such that

$$S_m > m \quad , \quad m < m_o \tag{i}$$

$$S_m \leq m \quad , \quad m \geq m_o \tag{ii}$$

Therefore $m=m_o$ uniquely satisfies (8), by noting that from (i) we have $m_o-1 < S_{m_o}-1$, or $m_o < 1+\sum_{i=o}^{m_o-1} Z^i$, which is the righthand side of (8), while from (ii) we have, $m_o \geq S_{m_o} = \sum_{i=1}^{mo} Z^i$, which is the lefthand side of (8). The solution is unique since for any $m'> m_o$ we have $m'-1> S_{m'-1}$ or $m'> 1 + \sum_{i=o}^{m'-1} Z^i$ and m' cannot satisfy the righthand side of (8). QED

Example: For $n=8$ and $p=.o1$, $.o15$, $.02$ (8) yields $m=5$, 4, 3 respectively, resulting in rates .1716, .2103 and .2520. In Table I these values are compared with the variable-to-fixed (VF) and variable-to-variable(VV) results of the full tree($n=8$, $k=o$).

4. Variable-to-variable runlength overfull tree encoding (k=1)

Consider the optimal overfull tree found by lemma 4. We can improve the rate by assigning variable lengths to the codewords according to Huffman's algorithm. It can be shown that the codeword-probabilities in this case satisfy (12),

$$P(A_i)=pq^{i-1} \sum_{j=1}^{m} \frac{Z^{j-1}}{j} \qquad ,i=1,2,\ldots n-2$$

$$P(A_n) = Z^m \tag{12}$$

$$P(A_{n-1}) = 1 - \sum_{\substack{i=1 \\ i\neq n-1}}^{n} P(A_i)$$

From Huffman's algorithm we then obtain the codeword-length vector $\underline{n}^d = \{n_1^d, n_2^d, \ldots, n_n^d\}$, where n_i^d is the length of the ith codeword represented in channel digits.
The length n_i of the ith message represented in channel digits equals,

$$n_i = \sum_{j=1}^{n} m_{ij} \, n_j^d \tag{13}$$

Where m_{ij} is the number of codewords A_j in the ith message. Note that (13) reduces to (4) in the case $\underline{n}^d=\{n_F, n_F, \ldots, n_F\}$. It can be shown that $\underline{n}$ satisfies (14),

$$\bar{n} = p\left(\frac{1-Z^m}{1-Z}\right)\left(\sum_{i=1}^{n-2} q^{i-1}\, n_i^d\right) + Z^m\, n_n^d + Z\left(\frac{1-Z^{m-1}}{1-Z} - (m-1)Z^{m-1}\right) n_{n-1}^d \qquad (14)$$

Formula (14) was used to find the variable-to-variable rates for k=1 in Table I.

Remark Note that we cannot use formula (5) to find R, by replacing n_F by $\bar{n}_d$, $\bar{n}_d = \sum_{i=1}^{n} P(A_i).n_i^d$ since $(1 + r)\, \bar{n}_d \neq \bar{n}$ in general. In the special case $\underline{n}^d=\{n_F, n_F, \dots n_F\}$ however, $(1+r).\,\bar{n}_d = (1+r).n_F=\bar{n}$.

The case k>1, in particular k=2

Some preliminary results have been obtained for k=2. We conjecture that the optimal runlength overfull encoding tree will always be a multiple concatenation of its full subtree T', i.e. n' = j(n-k-1)+1. More specifically, we conjecture that

$x_i \equiv o \bmod (n-k-1)$, $i= n-k, n-k+1,\dots,n$

where $\tilde{A}_i = 0^{x_i}$,i.e. $P(\tilde{A}_i)=q^{x_i}$.

A computer-search, carried out for the example in Table I for k=2, confirmed the conjecture. For n=8, p=.o1 we find $x_7=20$, $x_8=70$, while for p=.o15 and p=.02 we find $x_7=15$, $x_8=50$ and $x_7=15$, $x_8=40$ respectively. Based on the conjecture some expressions for the codeword probabilities and the rate of the code have been obtained. Since these expressions are rather complex and since we have not proved the conjecture, we have omitted here these formula's.

Bradley's code (n=8, p=.o5) also satisfies the conjecture, *viz.* $x_7=10$, $x_8=15$, but can be improved by the code with $x_7=10$, $x_8=25$, lowering the rate from .3662 to .3492, while h(.05)= .2864. Lemma 4 results in the (k=1)-code with m=3 and a rate .3685 and is apparently inferior to the optimal (k=2)-code.

5. Conclusions

Cohn's approach to the noiseless discrete source-coding problem, combined with Bradley's earlier runlength scheme have lead us to an interesting class of source-codes for the skewed binary source.
We have formulated the class of runlength overfull tree-codes and some characteristics for these codes are derived.
Numerous examples have shown that the overfull tree-codes for the skewed binary source are superior to the traditional full-tree codes (n'=n). Our intention is to generalize the results for non-binary and non-skewed sources. Also the case of more than one copious node needs further investigation.

References

1] B. D. Tunstall- Synthesis of Noiseless Compression Codes
PhD-thesis, Georgia Institute of Technology 1968
2] D. L. Cohn - Optimum Noiseless Source Codes with Fixed Dictionary Size. IEEE Int. Symp. on Information Theory, Ronneby, Sweden 1976.
3] S. D. Bradley - Optimizing a Scheme for Runlength Encoding
Proc. IEEE, vol. 57, pp. 108-109, Jan.'69
4] F. Jelinek, K.S. Schneider - On Variable-length-to-Block Coding
IEEE Trans. Inf. Theory, pp. 765-774, Nov. '72

5] L. Th. N. M Mulder, - On Variable-to-Variable Coding for Discrete
David L. Cohn Memoryless Sources.
To be presented at the IEEE Int. Symp. on
Information Theory, Ithaca, 1977

n	p	h(p)	R	R	n'	R	R	n'	R	n'
			k=o, VF	k=o, VV	k=o	k=1, VF	k=1, VV	k=1	k=2, VF	k=2
8	.010	.0808	.4416	.1757	8	.1716	.1642	31	.1322	71
8	.015	.1124	.4482	.1922	8	.2103	.2072	25	.1696	51
8	.020	.1414	.4550	.2086	8	.2520	.2246	19	.2018	41

Table I Comparison of the variable-to-fixed (VF) and variable-to-variable (VV) codes for k=o and k=1 and the variable-to-fixed codes for k=2, for n=8 and p=.o1, .o15 and .o2 respectively. n' denotes the number of messages in the code.

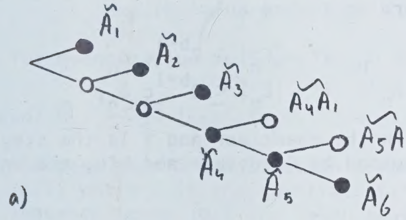
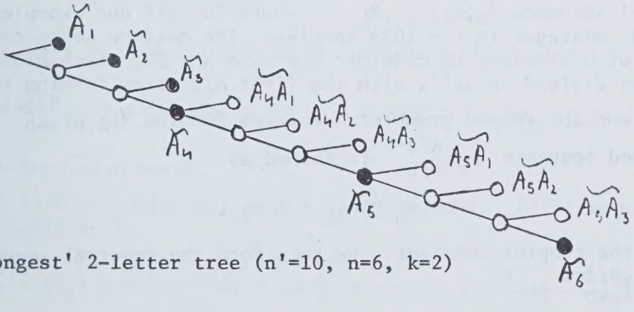

$\breve{A}_1$
$\breve{A}_2$
$\breve{A}_3$
$\overset{\frown}{A_4A_1}$
$\overset{\frown}{A_5A_1}$
$\breve{A}_4$
$\breve{A}_5$
$\breve{A}_6$
a)

Figure 6. a)'Shortest' 2-letter tree (n = n´ = 6, k = 2)


$\breve{A}_1$
$\breve{A}_2$
$\breve{A}_3$
$\overset{\frown}{A_4A_1}$
$\overset{\frown}{A_4A_2}$
$\overset{\frown}{A_4A_3}$
$\breve{A}_4$
$\overset{\frown}{A_5A_1}$
$\overset{\frown}{A_5A_2}$
$\overset{\frown}{A_5A_3}$
$\breve{A}_5$
$\breve{A}_6$

b) 'longest' 2-letter tree (n'=10, n=6, k=2)

COMPARISON OF SPC, PCM, AND ADPCM

NEAL C. GALLAGHER
School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

ABSTRACT

In this paper we investigate the effect of quantization in Spectral Phase Coding (SPC). We compare SPC to ADPCM and linear PCM in terms of signal to quantization noise ratio. Also, we discuss the nature of these quantization errors and how they appear in the decoded signal. Our results indicate that SPC performs well for a large class of signals and is relatively insensitive to signal variations.

I. Introduction

Adaptive Differential Pulse Code Modulation (ADPCM) is a technique used for digitizing a possibly large dynamic range sampled sequence; the ADPCM scheme takes advantage of sample to sample correlation and performs best whenever adjacent samples are highly correlated. Although adaptive by nature, certain parameter values in the ADPCM system must be designed to fit the signal inorder to achieve the best results [1,2]. Consequently, it may be difficult to accurately represent some signals using ADPCM. Reference 1 contains an excellent discussion of quantization errors in ADPCM and in Fig. 1 is found a block diagram of the ADPCM system as considered by Goldstein and Liu [1]. In Fig. 1, the term $\hat{X}_{n+1}$ represents the quantized version of the sample X_n. So, the quantizer input, denoted as E_n, is the difference between the present input sample and the quantized version of the previous sample. The adaptive quantizer has a step size of Δ_n for input X_n; this step size is updated according to the rule

$$\Delta_{n+1} = \begin{matrix} \gamma\,\Delta_n & \text{if} & |E_n| > 2^{b-1} c\, \Delta_n \\ \Delta_n/\gamma & \text{if} & |E_n| \le 2^{b-1} c\, \Delta_n, \end{matrix} \tag{1}$$

where b is the number of bits in the quantizer and γ is the step size adaptation parameter. As discussed by Goldstein and Liu, the value of c is limited to

$$c \in \left(\frac{1}{2^{b-2}}, \frac{2}{2^{b-2}}, \frac{3}{2^{b-2}}, \ldots, 2 - \frac{1}{2^{b-2}}\right); \tag{2}$$

the factor γ is restricted to values $1 < \gamma$.

The concept of Spectral Phase Coding (SPC) is substantually different from that of time-domain coding procedures such as ADPCM, this is because SPC is a transform coding scheme. When using the SPC method, we begin with a signal sequence $X_0, X_1, \ldots, X_L \ldots$, where for all our examples we truncate our messages to $L = 1024$ samples. The message sequence is divided into blocks of N samples; we consider the case $N = 32$. Each block of N terms is then divided in half, with the first N/2 terms forming the sequence $\{a_{1,n}\}_{n=0}^{N/2-1}$ and the second group of N/2 terms forming $\{a_{2,n}\}_{n=0}^{N/2-1}$. Then, the complex valued sequence $\{a_n\}_{n=0}^{N/2-1}$ is formed as

$$a_n = a_{1n} + i\, a_{2n},$$

where i is the complex constant. We then form the spectral sequence $\{A_p \exp(i\theta_p)\}_{p=0}^{N/2-1}$ as

$$A_p \exp(i\theta_p) = \sum_{n=0}^{N/2-1} a_n \exp(-i4\pi np/N) \quad p = 0,\ldots, \frac{N}{2}-1. \tag{3}$$

The SPC sequence $\{\psi_p\}$ is described by the following equations:

$$a_n = \frac{2}{N} \sum_{p=0}^{N/2-1} \frac{S}{2}\,[\exp(i\psi_p) + \exp(i\psi_{p+N/2})]\,\exp(i4\pi np/N), \quad n = 0,\ldots, \frac{N}{2} - 1 \tag{4}$$

where

$$S = \max_p \{A_p\} \tag{5a}$$

$$\psi_p = \theta_p + \cos^{-1}(A_p/S) \tag{5b}$$

$$\psi_{p+N/2} = \theta_p - \cos^{-1}(A_p/S),\ p = 0,.,\ldots, \frac{N}{2} - 1, \tag{5c}$$

where in the sequal we set $\phi_p = \cos^{-1}(A_p/S)$. Equation (5) describes the coding procedure and Eq. (4) the decoding procedure. For digital transmission, the values S, ψ_0, $\psi_1,\ldots,\psi_{N-1}$ must be quantized; the effect of quantizing the values $\{\psi_n\}$ is treated in Ref. 3. In section II of this paper we present a study of the effect of S-quantization on the reconstructed signal. Section III contains a number of examples by which we compare SPC to ADPCM and linear PCM.

II. Effect of S-Quantization

The ideal unquantized SPC representation is

$$A_p \exp(i\theta_p) = \frac{S}{2}\,[\exp(i\psi_p) + \exp(i\psi_{p+N/2})]; \tag{6}$$

accounting for quantization in S and $\{\psi_p\}$, we have

$$\hat{A}_p \exp(i\hat{\theta}_p) = \frac{S+n}{2}\,[\exp[i(\psi_p+\alpha_p)] + \exp[i(\psi_{p+N/2} + \alpha_{p+N/2})]], \tag{7}$$

where n is the zero mean S-quantization error; n is uniformly distributed over $(-\Delta/2, \Delta/2)$ where Δ is the quantizer step size. The phase quantization error sequence terms $\{\alpha_p\}$ are taken as independent random variables uniformly distributed over $(-\Theta/2, \Theta/2)$. We assume n to be independent of $\{\alpha_p\}$ and we call $\hat{A}_p \exp(i\hat{\theta}_p)$ the quantized version of $A_p \exp(i\theta_p)$. Forming the mean squared error between Eqs. (6) and (7), we have

$$\begin{aligned} \text{mse} &= E\{|A_p \exp(i\theta_p) - \hat{A}_p \exp(i\hat{\theta}_p)|^2\} \\ &= E\{\frac{S^2}{4}\,[\exp(i\phi_p)(1-\exp(i\alpha_p)) + \exp(-i\phi_p)(1-\exp(i\alpha_{p+N}))] \\ &\quad + \frac{n^2}{4}\,[2 + \exp[i(2\phi_p+\alpha_p-\alpha_{p/N})] + \exp[-i(2\phi_p+\alpha_p-\alpha_{p+N})]]\} \\ &\quad + \text{Zero mean terms}, \end{aligned} \tag{8}$$

where $E\{\cdot\}$ is the statistical expectation operator. After some algebra, Eq. (8) simplifies to

$$\text{mse} = S^2\left(1 - \frac{\sin\ \Theta/2}{\Theta/2}\right) + \frac{\sigma_n^2}{2}\left[1 + \cos(2\phi_p)\left(\frac{\sin\ \Theta/2}{\Theta/2}\right)^2\right], \tag{9}$$

where $\sigma_n^2 = E\{n^2\}$ and the value of S is treated as known. The first term in Eq. (9) represents the error due to phase quantization alone; the second term is the error due to the quantization of S. Also, by using the fact $\phi_p = \cos^{-1}(A_p/S)$, we have $\cos 2\phi_p = 2(A_p/S)^2-1$. Let X(n) be the time series reconstracted using the spectral terms in Eq. (6), and let $\hat{X}(n)$ be the time series reconstructed using Eq. (7); then, by use of Parseval's relation, we find

$$E\{|X_n - \hat{X}_n|^2\} = \frac{1}{N} E\{|A_p \exp(i\phi_p)-\hat{A}_p \exp(i\hat{\phi}_p)\}$$

$$= \frac{S^2}{N}\left(1 - \frac{\sin \theta/2}{\theta/2}\right)$$

$$+ \frac{\sigma_n^2}{2N}\left[1 - \left(\frac{\sin \theta/2}{\theta/2}\right)^2 \left(1 - \frac{2}{S^2} \sum_{n=0}^{N-1} X_n^2\right)\right] \qquad (10)$$

The following section contains a number of examples used for the comparison of SPC, ADPCM and linear PCM.

III. Examples

For our examples, we will use three different signals, each of which is a modification of the time series

$$X_{n+1} = 0.5\, X_n + Z_n, \qquad (11)$$

where $\{Z_n\}$ are independent random variables each with a uniform distribution over (-0.5, 0.5). Then, using the values X_n, we add a bias B_n to form $X_n + B_n$; we use the values of +0.5, -0.5, or 0.0 for B_n. The value employed changes at random instants in time. In this way, we generate the three signals illustrated in Figs. 2 through 4; we refer to these signals as S0, S1, and S2. The signal S0 is simply the unbiased sequence as generated by use of Eq. (11).

First, we discuss the performance of ADPCM when using the three test signals. The quantizer has 256 levels (b=8 bits); the initial quantizer range before adaptation begins is (-1, 1); the parameter Υ is set to Υ=2. Using the input S0, we perform computer simulations. These simulations are performed for different values of the parameter c. The measured signal to quantization noise ratio SNR are listed in Table 1.

c	0.015625	0.0625	0.125	0.5	0.78125
SNR (dB)	9.0	22.5	29.0	15.0	9.9

Table 1 Signal to Quantization Noise Ratios as a Function of C with S0 for ADPCM.

Thinking c = 0.125 to be a good value as indicated by Table 1, we perform another simulation for signal S1; we found a SNR of 20.3 dB. However, when using S1 and c = 0.015625, the simulation produced a SNR of 29.0 dB. These results clearly indicate that ADPCM performance depends greatly on the proper choice of c, or equivalently on the properties of the input signal. In our examples, the primary source of ADPCM error is occasional large variations in input signal amplitude and the resultant quantizer saturation.

We now consider the performance of SPC for these same three test signals; compared to ADPCM, SPC is relatively insensitive to signal parameter variations. We consider the case where the values $\{\psi_p\}$ are quantized to 256 levels over $(-\pi, \pi)$; the parameter S is also quantized to 256 levels over the range (0.0, 256.0). We note that this is very coarse S-quantization. To this SPC system we apply the three test signals S0, S1, and S2, and perform the SPC computer simulation; the respective SNR's are 22.8 dB, 33.7 dB and 33.9 dB. To illustrate the effect of S-quantization on the SNR, we perform the simulations a second time without S-quantization. The respective SNR's are 38.8 dB, 33.8 dB and 34 8 dB. Except for the signal S0, S-quantization has a negligible effect. To help understand the reason for this behavior, we list some typical values for S in Table 2. Also listed in Table 2 are those values to which S is quantized.

S	3.73	4.70	3.85	4.10	7.04	3.35
S_Q	4.02	5.02	4.02	4.02	7.03	3.01

Table 2 Typical Values of S and Their Quantized Versions S_Q as for Input S0.

For comparison, Table 3 contains typical values of S generated using signal S1.

S	114.47	113.89	114.49	113.01	106.48	114.59
S_Q	114.45	113.44	114.45	113.44	106.42	114.45

Table 3. Typical Values of S and S_Q Generated by Input S1.

Because the same quantization step size is employed in both of these cases, the S quantization error is much more significant for signal S0 than for signal S1.

In order to estimate the validity of our theoretical analysis as outlined in the previous section, we compute the SNR by use of Eq. (10), and compare these results with the simulation results. Specifically,

$$\mathrm{SNR} = -10 \log_{10} \left[\sum_{n=0}^{N-1} E\{ | X_n - \hat{X}_n |^2 \} / \sum_{n=0}^{N-1} X_n^2 \right] , \tag{12}$$

where N=32. The phase quantizer step is $\theta = 2\pi/256$ for a 256 level quantizer; the S quantization error variance is $\sigma_n^2 = 1/12$. When using signal S0, we replace $\sum_{n=0}^{N-1} X_n^2$ by $\sum_{n=0}^{64} E\{X_n^2\}$. With the use of Eq. (11), this sum is computed as 64/9. To obtain a typical value for S^2, we use the average of those values actually measured in the simulation; this becomes $S^2 = 18.6$. By direct substitution of these values into Eq. (10), we find

$$\mathrm{SNR} = -10 \log_{10} [3.280 \times 10^{-2}/3.556] = 20.4 \text{ dB}.$$

Similarly, for signal S1 we use $S^2 = 1.246 \times 10^4$ and $\sum_{n=0}^{N-1} E\{X_n^2\} \approx 800.0$; in this case Eq. (10) produces SNR = 34.0 dB. When using signal S2, we find SNR = 34.3 dB. These results agree very well with our simulations.

The effect of S-quantization is particularly noticable when the input is signal S0; however, no attempt has been made to optimize the S quantizer. Although performance is almost ideal when signals S1 and S2 are employed, even for suboptimum S-quantization. This result stands in contrast to ADPCM, where performance depends on parameter values c and γ. However, we also note that in our simulations, ADPCM outperformes SPC whenever a small number of quantization levels are used - say 8 levels or less.

Whereas ADPCM suffers quantizer saturation for large signal variations, PCM suffers quantizer saturation for large excursions in signal value; on the other hand, SPC is not particularly sensitive to either phenomina. As depicted by Eq. (4), the parameter S represents a scale factor in signal reconstruction; S-quantizer saturation means simply that the reconstructed signal is improperly scaled. Unlike in PCM, S-quantizer saturation does not destroy fine signal variations. However, our studies indicate that for moderate dynamic range signals, PCM performs very well. For example, an 8 bit linear PCM quantizer with 4σ loading has a SNR of about 40.9 dB for all signals; for signal S0, the linear quantizer step is $\Delta = 0.0104$; for S1, $\Delta = 0.1563$, and for signal S2, $\Delta = 0.0705$.

An important point to keep in mind is that signal to noise ratio is not always an effective measure of performance. For example, suppose that in signal S1, the information of interest is carried by the fine variations of the signal; in this situation, a quantizer step size of $\Delta = 0.1563$ is much too coarse. If signal to noise ratio is not always a valid measure of system performance, then how do we compare the relative performance of ADPCM, SPC, and PCM? The answer is that we must consider the nature of the errors for each separate method; PCM error appears as a loss in dynamic range, and system performance depends on step size. ADPCM error appears as an inability to track large first derivatives in the signal, and performance depends on quantizer step size. SPC retains the ability to track fine signal variations even with S-quantizer saturation; SPC performance seems relatively insensitive to S-quantizer step size.

The support of the National Science Foundation under grant ENG 76-82426 is gratefully acknowledged.

REFERENCES

1. L. H. Goldstein and B. Liu, "Quantization Noise in ADPCM Systems", IEEE Trans. on Comm., COM-25, 227 (1977).

2. L. H. Goldstein and B. Liu, "Quantization Error and Step-Size Distribution in ADPCM", IEEE Trans. on Info. Theory, IT-23, 216 (1977).

3. N. C. Gallagher, "Quantization Schemes for the DFT of a Random Time Series", To appear in IEEE Trans. on Info. Theory, January 1978.

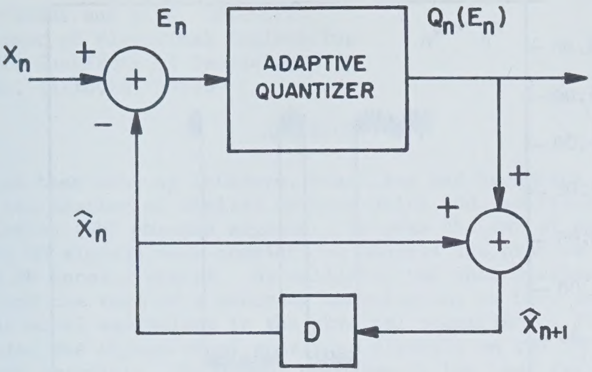

E_n
$Q_n(E_n)$
X_n
+
−
ADAPTIVE QUANTIZER
+
+
$\hat{X}_n$
+
D
$\hat{X}_{n+1}$

Fig. 1. ADPCM System with Input X_n.

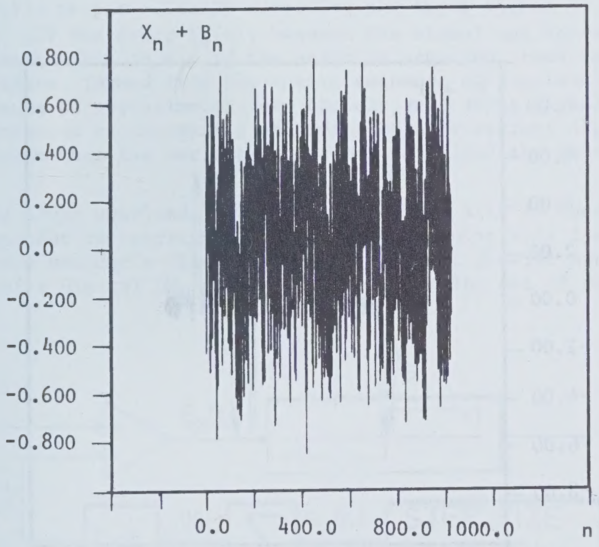


Fig. 2 Signal S0

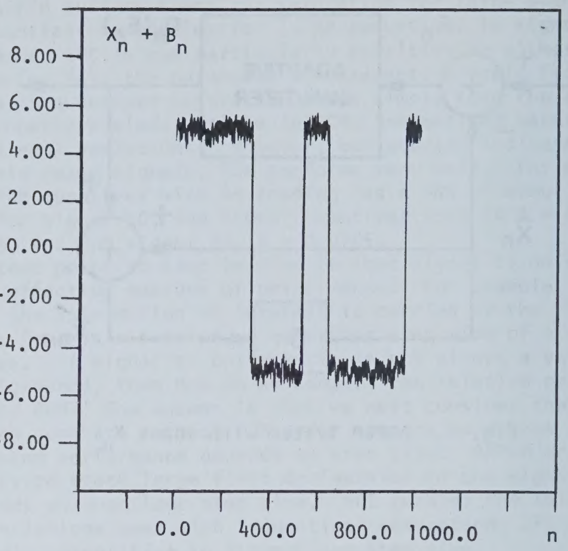


Fig. 3 Signal S1

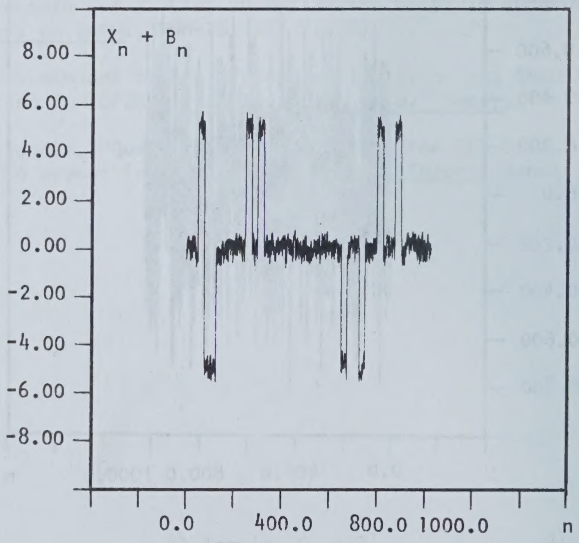


Fig. 4 Signal S2

DIGITAL FILTERING OF DELTA MODULATION ENCODED SIGNALS

D. P. PREZAS and J. L. LOCICERO
Department of Electrical Engineering
Illinois Institute of Technology
Chicago, Illinois 60616

ABSTRACT

Work has been done by LoCicero, Schilling and Gardonick [1,2] on the design and realization of digital devices which add, subtract or multiply delta modulation (DM) encoded signals. To make the set of operations dealing with DM signals more complete we address the problem of digitally filtering a DM encoded signal. By utilizing the characteristics of an all-digital DM and the form of a discrete convolution, we have developed a mathematical model equivalent to the physical operation of filtering. This model achieves the objective of operating directly on the DM bit stream without first decoding. The filter performance has been tested via simulation on a digital computer. The equivalent digital circuit for the above model has been designed, restricting ourselves to standard digital hardware. Although the development has been completed only for signals encoded with a linear digital DM, the general theory can easily be extended to include certain classes of adaptive DMs.

INTRODUCTION

Delta modulation is a technique, diagramatically shown in Fig. 1, by which an analog signal, $x(t)$, is encoded into a sequence of binary digits (bits), $\{e_x(k)\}$, by periodically comparing the input signal to an estimate signal, $\hat{x}(t)$. If the error $\xi_x(t)$, between the signal and the estimate is positive, then the bit is +1; if the error is negative, then the bit is -1. The estimate, formed from the entire sequence of previously generated DM bits, is made to approximate, very closely, the input signal, by appropriately increasing or decreasing according to a prescribed algorithm. The amount of change that the estimate undergoes is called the DM step size, $s_x(k)$.

To avoid slope overload, we have assumed that $x(t)$ is bandlimited to f_m and the encoder is operating well above the Nyquist rate i.e., $f_s >> 2f_m$, where f_s is the encoder's clock frequency. The most general mathematical description of a digital DM is given by the following set of equations:

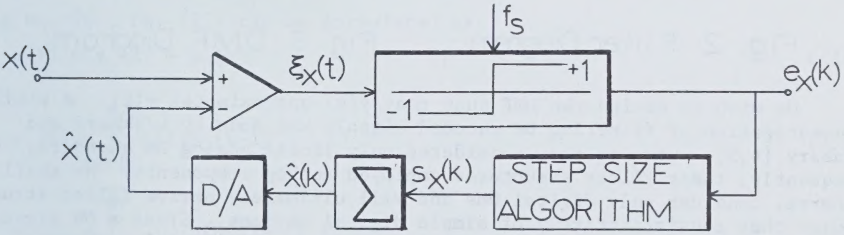


Fig. 1 Basic Digital DM

$$e_x(k) = \mathrm{sgn}[\xi_x(k)], \tag{1}$$

$$\xi_x(k) = x(k) - \hat{x}(k) \tag{2}$$

and

$$\hat{x}(k) = \hat{x}(k-1) + s_x(k), \tag{3}$$

where k denotes the k^{th} time interval.

The particular type of DM is characterized by the step size algorithm. For the linear DM,

$$s_x(k) = Se_x(k-1), \tag{4}$$

where S denotes the magnitude of the minimum step size. For an adaptive DM, the step size algorithm is such that the step size adapts itself to the input signal. A representative class of adaptive DMs is derived by minimizing a mean square cost function. An example of this type of adaptive DM is described by the Song Algorithm [3]. In this algorithm, the step size is a function of the two past DM bits and the previous step size. Since we shall restrict ourselves to encoding voice-type signals, the appropriate adaptive encoder is the Song audio mode DM, where the step size changes linearly, i.e.,

$$s_x(k) = |s_x(k-1)|e_x(k-1) + Se_x(k-2). \tag{5}$$

DM FILTER SPECIFICATION

Let $H_d(f)$ be a desired filter frequency response and h(k) the corresponding truncated, shifted, discrete impulse response such that:

$$h(i)=0, \qquad i<0 \quad \text{and} \quad i>N. \tag{6}$$

Also let H(f) be the frequency response corresponding to h(k). Figure 2 shows the general block diagram of a filter where x(k) is the input signal and y(k) the output signal. We shall assume, without loss of generality that

$$x(i)=0, \qquad i<0. \tag{7}$$

Figure 3 shows a system equivalent to Fig. 2, where the filter function, H(f), has been replaced by a set of operations, hereafter called the delta modulation filter (DMF), and the input, x(k), by its DM equivalent bit stream, $e_x(k)$.

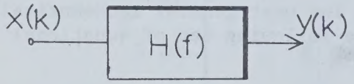


Fig. 2 Filter Diagram

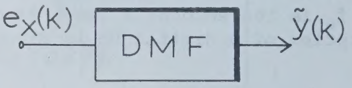


Fig. 3 DMF Diagram

We wish to design the DMF such that $\tilde{y}(k)$ approximates y(k). A similar investigation of filtering DM encoded signals was done by Lockhard and Babary [4,5]. However, they considered only linear analog DM encoders. Consequently, their filter structures employed analog components. We shall, of course, consider only digital DMs and will ultimately derive filter structures that consist entirely of simple digital devices. Since a DM signal consists of a sequence of ±1s, we will introduce a filtering technique requiring basic logical manipulations of these bits. We wish to avoid any complex digital operations requiring excessive hardware to realize the DMF.

The initial development of the DMF utilizes $\hat{h}(k)$, the DM estimate to h(k), corresponding to a low pass filter (LPF) so that it will be fairly easy to evaluate its performance. In this development, we assume that $\hat{h}(k)$ is symetric around N/2, i.e.,

$$\hat{h}(i) = \hat{h}(N-i), \qquad i>N/2. \tag{8}$$

This assumption does not effect the general approach; it does, however, slightly change the form of the final results, allowing us to use only N/2 samples of the entire $\hat{h}(k)$ instead of N samples. Since Fig. 2 is a linear, time-invariant, causal digital system, its output can be given by the following discrete convolution, i.e.,

$$y(k) = \sum_{i=0}^{\infty} h(i)x(k-i). \tag{9}$$

If we consider this discrete convolution as a model for the DMF shown in Fig. 3, Eq. (9) becomes:

$$\tilde{y}(k) = \sum_{i=0}^{\infty} \hat{h}(i)\hat{x}(k-i), \tag{10}$$

where we have used DM estimate signals instead of the actual discrete values for h(i) and x(i), and the approximation $\tilde{y}(k)$ instead of y(k).

For a general input signal, x(k), satisfying Eq. (7), we may rewrite Eq. (10), utilizing Eq. (6), as follows:

$$\begin{aligned}\tilde{y}(k) &= \sum_{i=0}^{k} \hat{h}(i)\hat{x}(k-i), \quad 0<k<N,\\ &= \sum_{i=0}^{N} \hat{h}(i)\hat{x}(k-i), \quad k\geq N.\end{aligned} \tag{11}$$

Here we shall consider only the case for $k \geq N$ since the development for $0<k<N$ is similar. Using DM properties [Eq. (3)] for $\hat{x}(k-i)$, Eq. (11) can be expressed as:

$$\tilde{y}(k) = \tilde{y}(k-1) + z(k), \tag{12}$$

where

$$z(k) = \sum_{i=0}^{N} \hat{h}(i)s_x(k-i). \tag{13}$$

Using Eq. (8), Eq. (13) can be formulated as:

$$z(k) = z_1(k) + z_2(k), \tag{14}$$

where

$$z_1(k) = \sum_{i=0}^{N/2} \hat{h}(i)s_x(k-i) \tag{15}$$

and

$$z_2(k) = \sum_{i=N/2+1}^{N} \hat{h}(N-i)s_x(k-i). \tag{16}$$

Employing the change of variables j = k-i and DM properties [Eq. (3)] for $\hat{h}(i)$, Eq. (15) eventually takes the form

$$z_1(k) = z_1(k-1) - \hat{h}(N/2)s_x(k-N/2-1) + \sum_{j=k-N/2}^{k} s_h(k-j)s_x(j). \qquad (17)$$

Similarly Eq. (16) is restructured as:

$$z_2(k) = z_2(k-1) + \hat{h}(N/2)s_x(k-N/2-1) - \sum_{j=k-N}^{k-N/2-1} s_h(N-k+j+1)s_x(j)$$
$$- \hat{h}(0)s_x(k-N-1). \qquad (18)$$

We now see that the sum of $z_1(k)$ and $z_2(k)$ yields a recursive relationship for the variable $z(k)$. It is exactly this form that permits us to easily realize the DMF with simple digital hardware. The general design of the DMF thus consists of the following set of equations:

$$\tilde{y}(k) = \tilde{y}(k-1) + z(k) \qquad (19)$$

and

$$z(k) = z(k-1) + \sum_{i=0}^{N/2} s_h(i)s_x(k-i) - \sum_{i=N/2+1}^{N} s_h(N+1-i)s_x(k-i)$$
$$- \hat{h}(0)s_x(k-N-1), \qquad (20)$$

where the change of variables $i = k-j$ has been used to obtain the final form of Eq. (20).

If we consider the case of a linear digital DM and apply the step size algorithm given in Eq. (4) for $s_h(k)$ and $s_x(k)$, Eq. (20) simplifies to become:

$$z(k) = z(k-1) + S^2 \sum_{i=0}^{N/2} e_h(i-1)e_x(k-i-1) - S^2 \sum_{i=N/2+1}^{N} e_h(N-i)e_x(k-i-1)$$
$$-\hat{h}(0)Se_x(k-N-2). \qquad (21)$$

The pair of equations (19) and (21) constitutes the DMF for a linear digital DM when $k \geq N$ and the system can be realized using simple digital components since $e_x(k)$ and $e_h(k)$ only take on values of ±1. The $\{e_x(k)\}$ constitutes the incoming bit stream, while the $\{e_h(k)\}$ is evaluated via a DM process given the impulse response $h(k)$ and is a set of stored constants for a specific filter design. Similar results for the cases $0<k\leq N/2$ and $N/2<k<N$ are given by the pairs of equations (22), (23) and (24), (25) as follows:

$$\tilde{y}(k) = \tilde{y}(k-1) + z(k) + \hat{h}(k)\hat{x}(-1), \qquad (22)$$

$$z(k) = z(k-1) + S^2 \sum_{i=0}^{k} e_h(i-1)e_x(k-i-1), \quad 0<k\leq N/2, \qquad (23)$$

and

$$\tilde{y}(k) = \tilde{y}(k-1) + z(k) + \hat{h}(N-k)\hat{x}(-1), \qquad (24)$$

$$z(k) = z(k-1) + S^2 \sum_{i=0}^{N/2} e_h(i-1)e_n(k-i-1) - S^2 \sum_{i=N/2+1}^{k} e_h(N-i)e_x(k-i-1),$$
$$N/2<k<N. \qquad (25)$$

In all of the previous results, we have assumed that $\hat{h}(-1) = 0$ and we shall assume that $\hat{x}(-1) = \pm S/2$.

DMF CIRCUIT DESIGN

To avoid hardware complexity in the realization of Eqs. (22) thru (25), where $0<k<N$ (since k, which is a running index, is also the upper limit of a sum) we assume that:

$$x(t) = 0, \qquad 0<k<N. \tag{26}$$

Equation (26) specifies the first N bits of the DM representation of the input signal as an alternating sequence of +1, -1 or -1, +1. Knowing this, we are able, using Eqs. (22) thru (25), to evaluate $z(N-1)$ and $\tilde{y}(N-1)$ in order to initialize the circuit. This will facilitate the realization of Eqs. (19) and (21).

Figure 4, on next page, shows the circuit diagram which realizes the pair of Eqs. (19) and (21). In order to include the period $0<k<N$, so that the circuit will be an exact representative of the DMF, we must initialize at time $k = N - 1$ at three points in the circuit. The symbols appearing in the circuit diagram of Fig. 4 are defined as follows:

d : a single-bit D flip flop with clock frequency Nf_s.
D : a single-bit D flip flop with clock frequency f_s.
FF : a multiple-bit D flip flop with clock frequency as indicated.
All gates indicated are single-bit AND gates.
The summers are multiple-bit full adders.

TESTING AND PERFORMANCE OF THE DMF MODEL

We applied our results for the case when $H_d(f)$ is an ideal LPF with cutoff frequency f_c = 3K Hz. The clock frequency of the DM was set at f_s = 120K Hz and the truncated $h(k)$ contained a total $N = 80$ points. The filter frequency response, $H(f)$, corresponding to the discrete impulse response $h(k)$, is shown in Figs. 5 and 6.

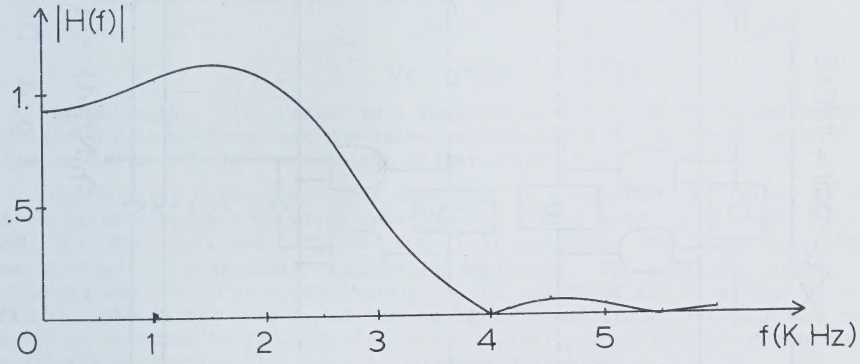


Fig. 5 Magnitude of H(f)

The selection of the magnitude of the minimum step size, S, was done on the basis of minimizing the error function

$$E(S) = \sum_{i=0}^{N/2} \left|\hat{h}(i,S) - h(i)\right|^2 \tag{27}$$

The minimization shows that the function E(S) exhibits a fairly flat portion around a minimum as indicated in Fig. 7. The value of S for which the minimum of E(S) occurs is S = 0.33 Volts. For values of S greater than

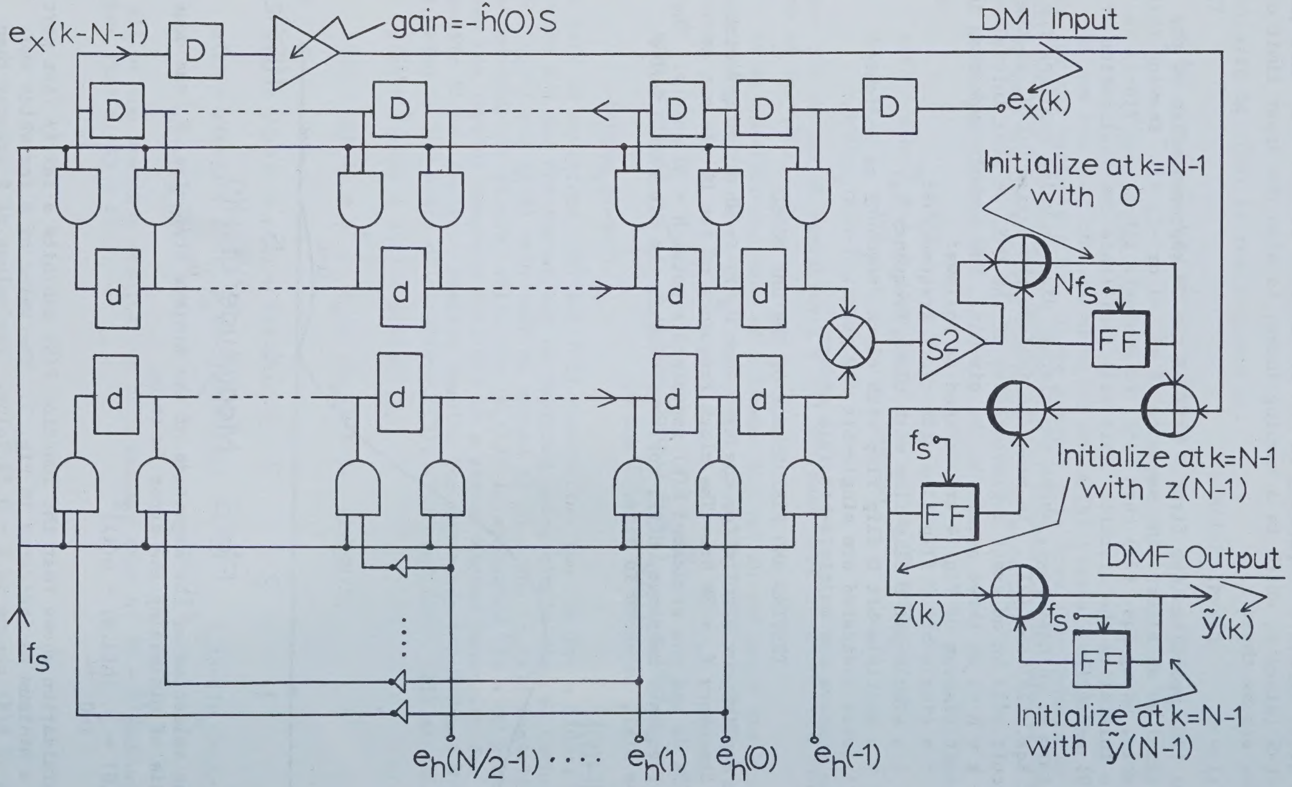

$e_x(k-N-1)$
D
gain=$-\hat{h}(0)S$
DM Input
D
D
D
D
D
$e_x(k)$
Initialize at k=N-1
with 0
d
d
d
d
$N f_s$
FF
S^2
d
d
d
d
f_s
FF
Initialize at k=N-1
with z(N-1)
DMF Output
z(k)
f_s
FF
$\tilde{y}(k)$
f_s
Initialize at k=N-1
with $\tilde{y}(N-1)$
$e_h(N/2-1)$ · · · ·
$e_h(1)$
$e_h(0)$
$e_h(-1)$

Fig. 4 DMF Circuit Diagram

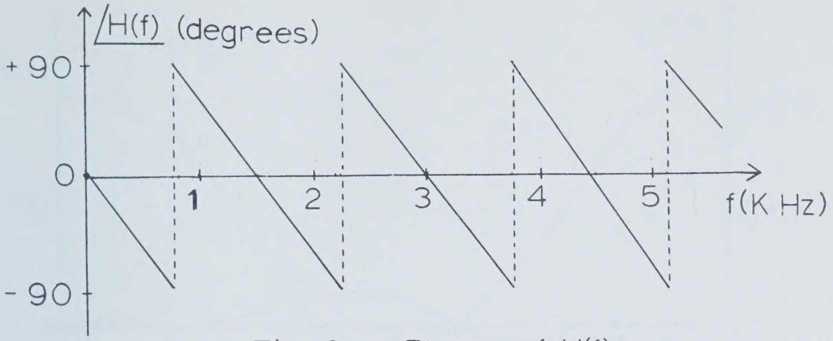


Fig. 6 Phase of H(f)

0.40 V or less than 0.26 V, E(S) increases dramatically. In our digital computer simulation, we used the value S = 0.28 Volts. We also used this same value of minimum step size for the DM encoding of both h(k) and x(k).

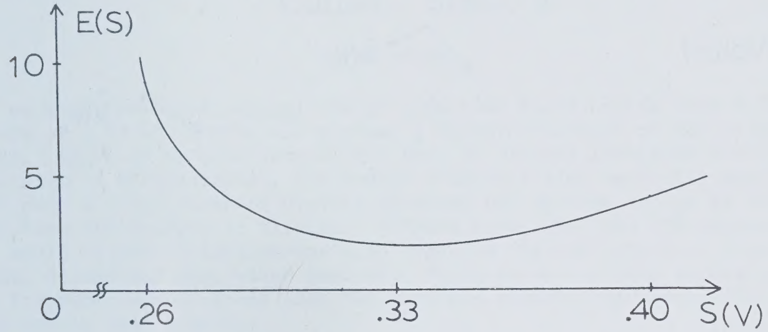


Fig. 7 Variation of E(S)

Assuming that x(k) varies in a range from -5 V to +5 V, we appropriately scaled h(k) so that its maximum value was exactly 5 V. As a consequence of this we had to inversely scale the filter output.

Our initial performance test consisted on a computer simulation of the three pairs of equations which describe the filter model, i.e., Eqs. (19) and (21), Eqs. (22) and (23), and Eqs. (24) and (25). The first test signal was a single tone sinusoid of constant amplitude. The frequency of the sinusoid was set to be approximately in the middle of the pass-band of the filter. The filter output had the appropriate amplitude,frequency and time delay as dictated by the transfer function,H(f). This indicated that our model was performing the desired filtering operation.

The next test input was a trapezoidal pulse. The slope of the linear portions of the trapezoid was set at $\pm S/T_s$, where $T_s = 1/f_s$. This represents the maximum slope that a linear DM can follow without slope overloading. The frequency spectrum of this input is indicated in Fig. 8.

The response of the DMF to the trapezoidal input signal is shown in Fig. 9. In this plot we also show the DM encoded input $\hat{x}(k)$, obtained by directly decoding $\{e_x(k)\}$. We observe that the effect of the LPF operation is to eliminate the sharp corners of the trapezoid. We also see that the time delay is approximately $40T_s$, agreeing with the linear phase

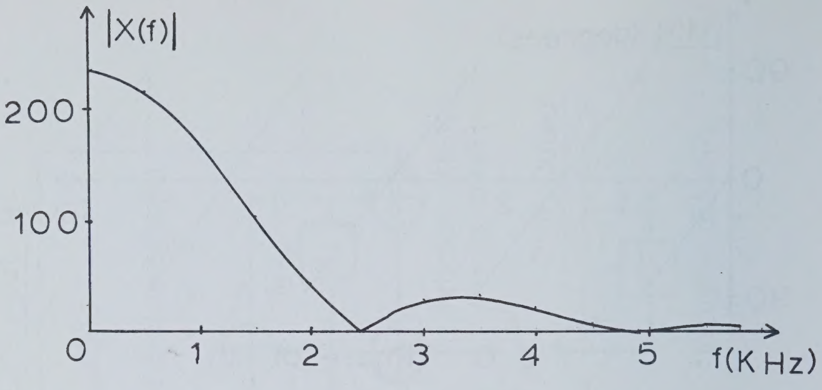


Fig. 8 Spectrum of a Trapezoidal Signal

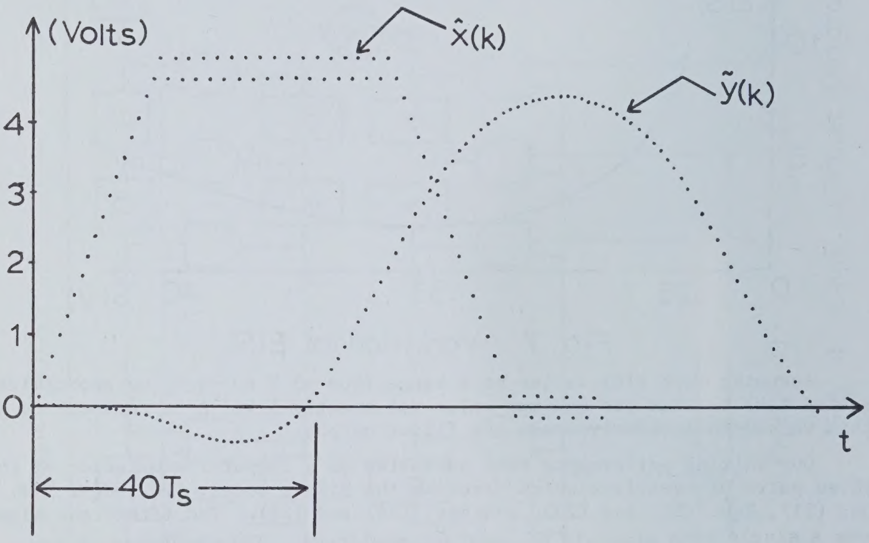


Fig. 9 Input and Output Pulse

characteristics of H(f). In Fig. 10, we display the amplitude spectra of Y(f) and $\tilde{Y}(f)$, where $Y(f) = X(f)H(f)$ and $\tilde{Y}(f)$ is the spectrum of $\tilde{y}(k)$. This graph shows that the shape of $|Y(f)|$ is approximated quite well by $|\tilde{Y}(f)|$ except for an attenuation in the passband. These results further verify the operation of the DMF.

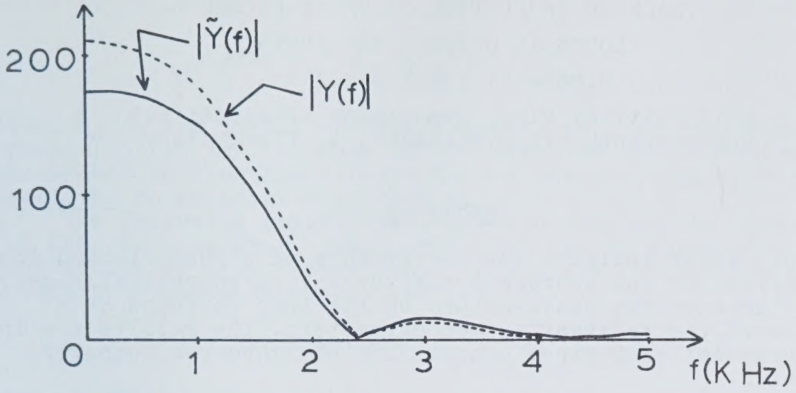


Fig. 10 Expected and Experimental Output Spectra

CONCLUSION

We have presented general design equations for a system that will operate on a DM bit stream and produce a filtered version of the original signal. Although we have pursued the case of digital linear DM coded signals and a LPF operation, our design equations also apply for signals coded with a large class of digital adaptive DMs and can easily be modified to include other types of filters. We have shown that the DMF design equation could be easily implemented with standard digital hardware, that is, adders, delays and hard-wired scalars. Furthermore, we have demonstrated that its performance agrees with the expected response of a specified LPF for a set of test signals.

An extension of this work is to investigate the hardware structure of a DMF for adaptive DM coded signals and several other types of filter functions. The advantage of using adaptive DM coded signals is that the bit rate is lower than for linear DM coded signals. This will result in fewer delay units in our DMF circuit and thus less total hardware.

REFERENCES

[1] J. L. LoCicero, D.L. Schilling and J. Garodnick, "Direct arithmetic processing of delta modulation encoded signals", IEEE Nat. Telecomm. Conf., pp. 392-397, Dec. 1974.

[2] J. L. LoCicero, J. Garodnick and D. L. Schilling, "The realization of arithmetic processors for delta modulation encoded signals", IEEE Nat. Telecomm. Conf., pp. 34-1 - 34-5, Dec. 1975.

[3] C.L. Song, J. Garodnick and D. L. Schilling, "A Variable-Step-Size Robust Delta Modulator", IEEE Trans. Commun. Tech. vol. COM-19, pp. 1033-1044, Dec. 1971.

[4] G. B. Lockhart, "Digital encoding and filtering using delta modulation", Rad. and Elec. Engnr., 42, No. 12, pp 547-551, Dec. 1972.

[5] G. B. Lockhart and S. P. Babary, "Binary transversal filters using recirculating shift registers", Rad. and Elec. Engnr., 43, No. 3, pp. 224-226, Mar. 1973.

TRACKING PROPERTIES OF PHASE-LOCKED LOOPS IN OPTICAL RECEIVERS

U.MENGALI and E.PEZZANI

University of Pisa, Department of Electrical
Engineering, Via Diotisalvi, 2, Pisa, Italy

ABSTRACT

This paper analyzes the performance of a phase-locked loop (PLL) driven by the photodetector current in an optical receiver. The variance of the phase-errors of the loop is found as a function of the relevant system parameters. The results are in good agreement with experimental data obtained via computer simulation.

1.INTRODUCTION

In recent years considerable research work has been carried out on the behavior of PLL's for synchronous communication applications over additive Gaussian channels [1]-[4]. As a result, at present, a satisfactory theory exists for evaluating and testing the performance of these synchronization circuits.

More recently, as a consequence of the increasing interest in optical communications, the problem arose of adapting the previous results to the case of optical channels. As known, the novelty here is the shot noise which, contrary to the Gaussian noise, is not additive and is statistically dependent on the signal. This gives another dimension of complexity to the synchronization problem so that it is felt that much work is still necessary to bring the optical synchronization theory to the same level of maturity it has reached with traditional channels.

Interesting work in this area is due to Gagliardi and Haney [5] who studied the behavior of a PLL when the timing information is transmitted by sinusoidally modulating the optical power at a frequency equal to the transmission rate. Here, we shall extend their results to the case in which the optical power is intensity modulated by a pulse amplitude modulated (PAM) signal and the timing information is directly extracted from the corresponding current generated in the photodetector of the receiver.

2.SIGNAL AND NOISE AT THE PLL INPUT

Following [7] , we write the photodetector output current as

$$i(t)=n_G(t)+\sum_k g_k w(t-t_k) \qquad (1)$$

where $\{g_k\}$ are the gains in the avalanche-detector mechanism, w(t) is the impulse response of the photodetector circuitry, $n_G(t)$ is Gaussian noise and $\{t_k\}$ are the photon arrival times on the detector surface. They are Poisson distributed with an intensity function

$$\lambda(t)=\lambda_o+\sum_k c_k p_o(t-kT-\tau) \qquad (2)$$

Here, λ_o is the so-called dark current, $\{c_k\}$ are the transmitted symbols, $p_o(t)$ is the shape of the received optical pulse, T is the symbol spacing and finally τ is a unknown, constant timing parameter to be tracked by the PLL.

The following remarks will prove useful in the analysis of the PLL. By elementary calculations it can be shown that the expectation of i(t), conditioned on a given realization of the intensity process $\lambda(t)$, has the form:

$$E\{i(t)\,|\,\lambda(t)\}=\text{constant}+G_1\sum_k c_k p_1(t-kT-\tau) \qquad (3)$$

where $G_n=E\{g_k^n\}$ and $p_1(t)$ is the convolution of $p_o(t)$ with w(t). From (3) we see that $E\{i(t)\,|\,\lambda(t)\}$ represents the transmitted PAM signal.

We further note that, letting $C_n=E\{c_k^n\}$, the unconditional expectation of i(t) becomes:

$$E\{i(t)\}=\text{constant}+G_1C_1\sum_k p_1(t-kT-\tau) \qquad (4)$$

i.e., a periodic function of time, with period T. This function can then be expanded in a Fourier series and in doing so we find that the fundamental term in the series can be written as $I_1\sin(w_st+\theta_1)$, where $w_s=2\pi/T$ and

$$I_1=\frac{2G_1C_1}{T}\left|P_o\left(\frac{1}{T}\right)W\left(\frac{1}{T}\right)\right| \qquad (5)$$

$$\theta_1=\arg\left\{P_o\left(\frac{1}{T}\right)W\left(\frac{1}{T}\right)\right\}+\frac{\pi}{2}-\frac{2\pi\tau}{T} \qquad (6)$$

In (5)-(6) $P_o(f)$ and W(f) are the Fourier transforms of $p_o(t)$ and w(t).

Going back to (1) we now split the photodetector current into the sum of four terms:

$$i(t)=s(t)+n_G(t)+n_p(t)+n_s(t) \qquad (7)$$

where $n_G(t)$ is the same as in (1) and

$$s(t)=E\{i(t)\} \qquad (8)$$

$$n_p(t)=E\{i(t)\,|\,\lambda(t)\}-E\{i(t)\} \qquad (9)$$

$$n_s(t)=\sum_k g_k w(t-t_k)-E\{i(t)\,|\,\lambda(t)\} \qquad (10)$$

It is easily seen that whereas s(t) is a deterministic function, the other terms on the right of (7) are random functions. In the sequel s(t) will be referred to as timing wave since it bears the information on the timing parameter τ. As for $n_p(t)$ and $n_s(t)$, the first will be called pattern noise because it represents deviations from the timing wave due to the data pattern, the second shot noise, owing to its impulsive character (see (10)). In view of (7), we then look at s(t) as the synchronizing signal to be tracked by the PLL in the presence of Gaussian, pattern and shot noise.

3.PLL ANALYSIS

Following [3] it can be shown that the PLL depicted in fig. 1 is governed by the stochastic differential equation:

$$\frac{d\varphi}{dt} = \Omega - KF(p)\{A\sin\varphi + n(t)\cos(w_s t + \theta_1 - \varphi)\} \qquad (11)$$

In writing (11) the following notations have been used: $\varphi(t)$ is the phase error of the loop (i.e., the difference between the phase $w_s t+\theta_1$ of the fundamental harmonic of s(t) and the phase $w_o t+\hat{\theta}_1$ of the sinusoid produced by the voltage controlled oscillator (VCO)), Ω is the loop detuning (i.e., $\Omega = w_s - w_o$), K is the gain of the VCO, F(p) is the transfer function of the loop filter and p=d/dt is the Heaviside operator, A is equal to $I_1/2$ (see (5)), θ_1 is defined in (6), and finally n(t) is the overall noise

$$n(t) = n_G(t) + n_p(t) + n_s(t) \qquad (12)$$

The problem of studying the statistical properties of the solution $\varphi(t)$ of (11) is very complicated. Essentially, two approaches can be followed: one is based on the solution of an appropriate kinetic equation [5]-[6] , the other makes use of perturbation techniques [8]. In the sequel we shall report on the results of this second method.

For the sake of simplicity let us assume zero loop detuning. Then, in the steady-state, it is found that $\varphi(t)$ can be put in the form of a series: $\varphi(t) = \varphi_1(t) + \varphi_2(t) + \ldots$; where:

$$\begin{aligned} &\varphi_1(t) = H(p)\{n(t)\cos(w_s t + \theta_1)\} && (13) \\ &\varphi_2(t) = H(p)\{\varphi_1(t)n(t)\sin(w_s t + \theta_1) && \\ &\varphi_3(t) = H(p)\{-\frac{A}{3!}\varphi_1^3(t) - \frac{1}{2}\varphi_1^2(t)n(t)\cos(w_s t + \theta_1) && \\ &\qquad\qquad + \varphi_2(t)n(t)\sin(w_s t + \theta_1)\} && \end{aligned}$$

... and so on. The operator H(p) is defined as

$$H(p) = -\frac{KF(p)}{p + AKF(p)} \qquad (14)$$

Under the hypothesis of a "low" noise level, we can neglected all but the first term in the series. Physically, this amounts to assume a sufficiently high signal-to-noise ratio in the loop so that the fluctuations of $\varphi(t)$ are very small. Such a condition is necessarily satisfied in the majority of the practical applications.

Letting $\varphi(t) \simeq \varphi_1(t)$, from (13) we see that our problem is reduced to a linear problem and therefore the computation of the variance σ^2 of $\varphi(t)$ can be obtained by standard methods. The interested reader is referred to [8] for a complete report on the general results of this computation. Here we shall limit ourselves to the important case wherein the noise equivalent bandwidth B_L of the PLL is much smaller than the signaling rate 1/T, and the dark current together with the Gaussian noise are negligible. Under these hypotheses it is found:

$$\sigma^2 = \frac{G_2 B_L T}{G_1^2 C_1} \quad \frac{P_o(o) - |P_o(2/T)| \cos\gamma}{|P_o(1/T)|^2} \tag{15}$$

with

$$\gamma = 2\arg\{P_o(1/T)\} - \arg\{P_o(2/T)\}$$

Examination of (15) reveals that, for a fixed noise bandwidth, σ^2 is independent of the transfer function W(f) of the photodetector circuitry. We also see that σ^2 decreases by increasing C_1. Of course, this is physically intuitive since C_1 is proportional to the average received optical power. Finally we note that σ^2 vanishes when

$$P_o(o) = |P_o(2/T)| \cos\gamma$$

Such a condition is practically satisfied when the optical pulse $p_o(t)$ is very short with respect to the symbol spacing. This corresponds to receiving narrow bursts of light and we intuitively expect such optical signals to indeed offer unique synchronization features.

4. SIMULATION RESULTS

The theory developed above has been thoroughly checked by means of computer simulations. Here we report on these simulations and we compare them with the theoretical results.

The following model was assumed. The shape of the optical pulse was rectangular with unit area and duty cycle d. The data $\{c_k\}$ were binary (0 and $2C_1$), independent and equally likely. The photodetector circuitry and the loop filter were RC low-pass filters with time constants T_W and T_F. The dark current and the Gaussian noise were neglected, and the avalanche gains were held constant and equal to G.

Figures 2 and 3 give a sample of the results. As we can see, the theory is quite accurate until the phase-errors are small. For $\sigma^2 > 10^{-2}$, the first-order approximation is no longer valid and higher order terms should be taken into account in the series expansion of $\varphi(t)$.

Fig. 4 shows that, in the linear range of operation of the PLL, the probability density function of $\varphi(t)$ can be accurately approximated by a Gaussian curve. Here, in fact, we have drawn a Gaussian function with the same variance of $\varphi(t)$. Superimposed are simulation points obtained from $2\ 10^4$ samples of $\varphi(t)$.

In fig. 5 an analogous comparison is made. Here, however, the variance of the phase errors is chosen purposely large so as to forbid linear operating conditions. As we can see, in this case the experimental curve suggested by the simulation points appears more piked than the Gaussian curve.

5. CONCLUSIONS

A theory has been presented which describes and predicts fairly well the behavior of a PLL driven by the photodetector current in a optical communication receiver.

The main restriction in the analysis is that the signal-to-noise ratio in the loop is assumed sufficiently high so as to allow linear operating conditions of the PLL. Such conditions are usually met in a practical receiver because the timing information it needs must be quite accurate to avoid unaccetable penalities in terms of error probability.

On the other hand, when better approximations are required, higher-order terms can be retained in the series expansion of $\varphi(t)$. This should not entail such great complications if we are content with the second-third approximation.

REFERENCES

[1] A.J.VITERBI, Principles of Coherent Communications, McGraw-Hill (1966)
[2] J.J.STIFFLER, Theory of Synchronous Communications, Prentice Hall (1971)
[3] W.C.LINDSEY, Synchronization Systems in Communication and Control, Prentice-Hall (1972)
[4] W.C.LINDSEY and M.K.SIMON, Telecommunication System Engineering, Prentice-Hall (1973)
[5] R.GAGLIARDI and M.HANEY, USCEE Report 396, Dept.Elec.Eng., Univ.South Calif. (1970)
[6] R.GAGLIARDI and S.KARP, Optical Communications, Wiley (1976)
[7] S.D.PERSONICK, Bell Syst.Tech.J., 52, 843-874, (1973)
[8] U.MENGALI and E.PEZZANI, CSMDRT Report, Univ.Pisa, (1977)

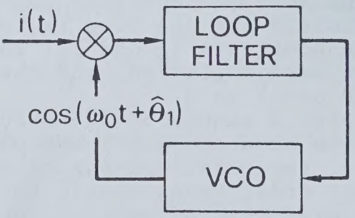


Fig. 1: PLL block diagram

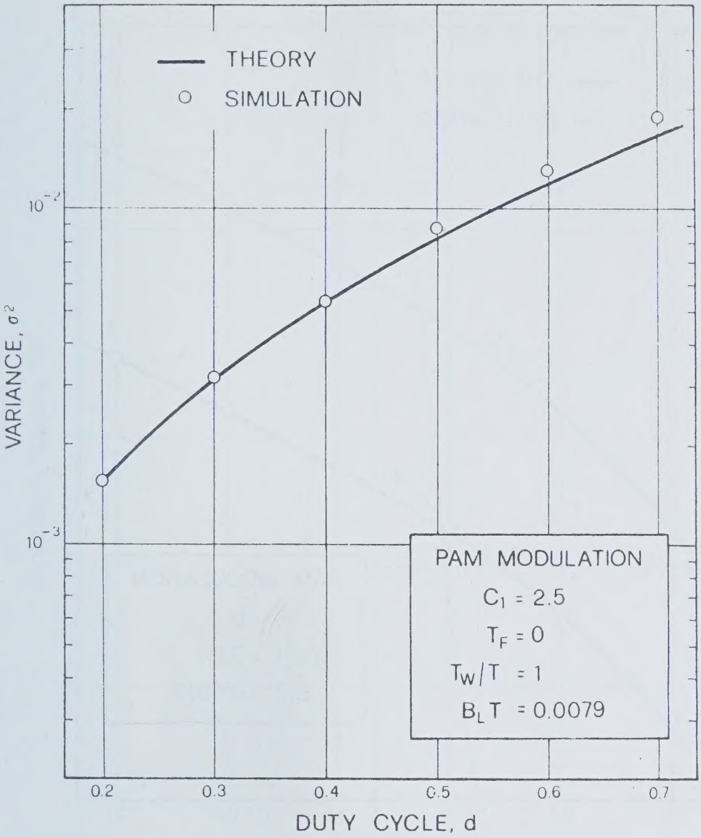


Fig. 2: Phase-error variance versus duty cycle

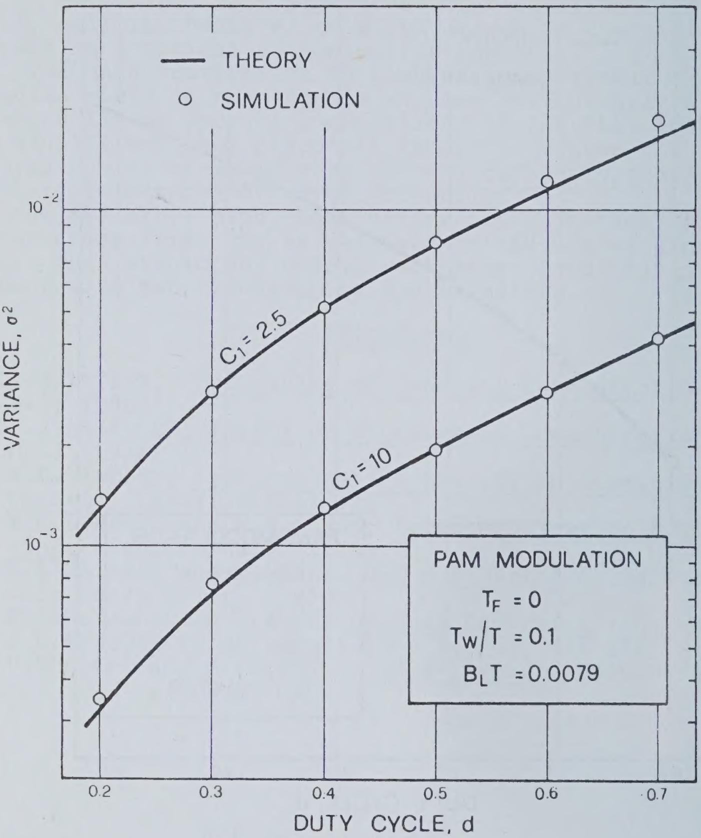


Fig. 3: Phase-error variance versus duty cycle, with C_1 as a parameter

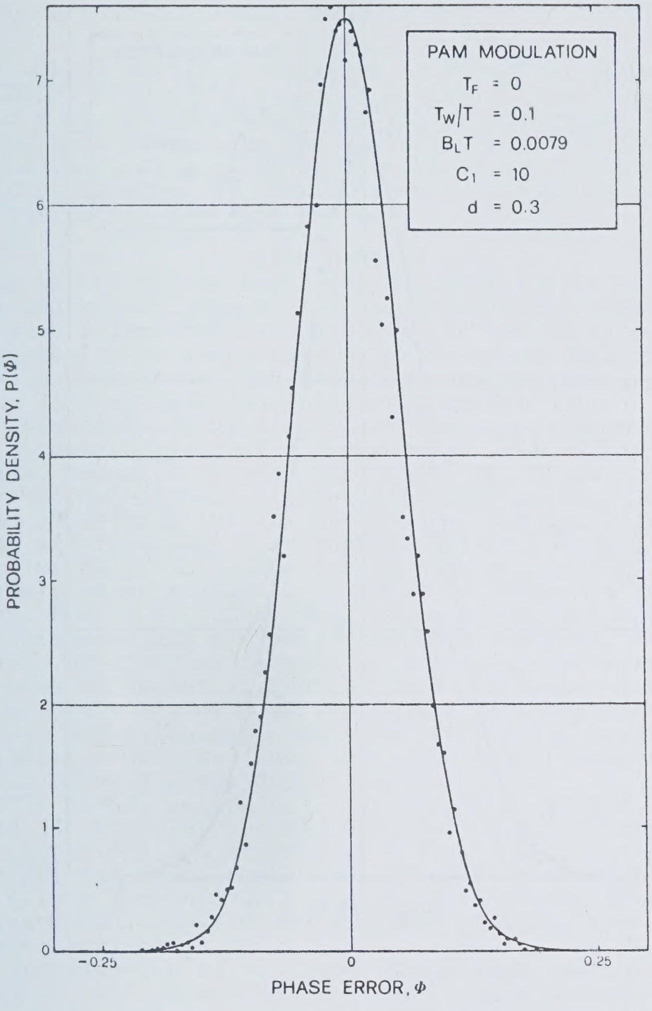


Fig. 4: Probability density function of the phase-errors

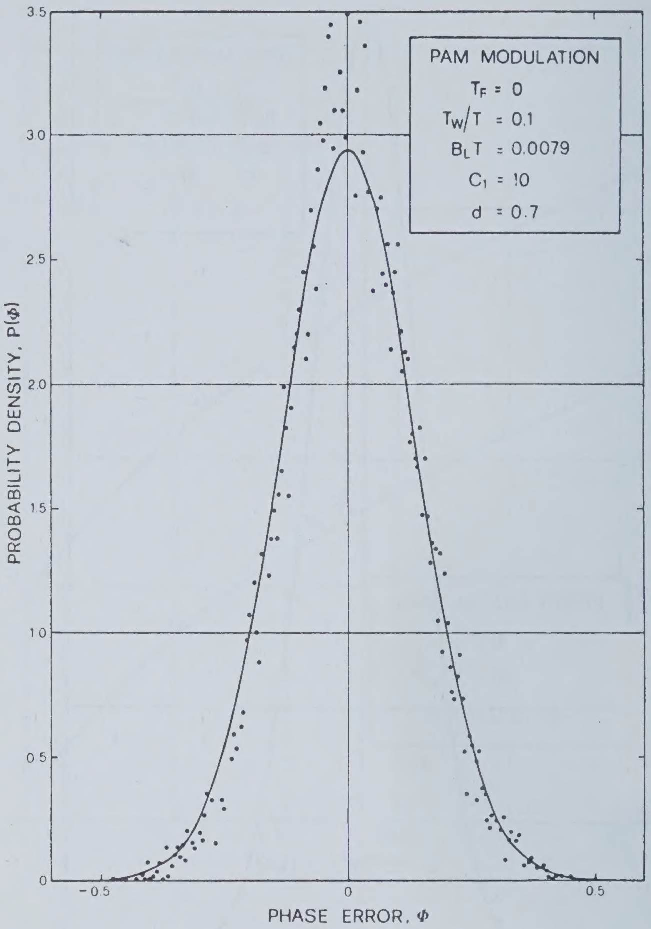


Fig. 5: Probability density function of the phase-errors

ON A MULTILEVEL QUANTIZED DIGITAL PHASE-LOCKED LOOP

FRANCO RUSSO* and NUNZIO ALDO D'ANDREA**
*CSMDRT-National Research Council, Pisa, Italy
**University of Pisa, Pisa, Italy

Abstract-The acquisition behavior of a Digital Phase-Locked Loop (DPLL), with a multilevel quantizer, is analyzed by means of a new graphical method. The existence and the stability of both stationary and periodic solutions are investigated and the cycles slipping phenomenon is displayed. Moreover, design constraints and a closed-form expression of the phase acquisition time are derived.

INTRODUCTION

Owing to the recent trend towards digital processing of signals and to the progress in large-scale integrated circuits, much effort has been directed to the development of digital phase-locked loops (DPLLs) (1). Various DPLLs have been reported in literature for symbol synchronization, carrier tracking or for angle demodulation purposes (1).

The present paper will consider the acquisition behavior, in the noiseless case, of a first-order DPLL with a multilevel quantizer, proposed by Reddy and Gupta (2). These authors have derived the non-linear, first-order, autonomous, difference equation describing the noise-free loop operation for the case of phase and frequency step inputs. Unfortunately a closed-form solution of this equation is unknown, so the loop has been simulated on a digital computer to study the acquisition behavior. But by so doing, some correlations between the system performances and system parameters are missed. Moreover, this means of studying non-linear difference systems may give only a limited amount of insight into the system behavior because, with the aim of discovering all possible non-linear phenomena, the calculation must be carried out for all possible initial conditions. This is a prohibitive task and will be of little value for the device design.

In this paper, making use of a graphical method, proposed by Russo (3) to solve a class of non-linear, autonomous difference equations, we examine in a simple and systematic manner the existence and the stability of both stationary and periodic (limit cycles) solutions, as well as the cycle slipping phenomenon. The aspects of these problems have not been fully investigated in (2).

The procedure, here outlined, shows that the condition for the acquisition, as derived in (2), is only necessary and suggests a sufficient condition to avoid cycle slipping, i.e. it allows one to obtain some design constraints on the relevant loop parameters. Moreover, other constraints are derived in order to assure that the final phase error be restricted in a prefixed range and to obtain a faster acquisition.

Finally, a closed-form expression of the phase acquisition time is derived when the loop relevant parameters are chosen according to the abovementioned constraints.

It should be noted that the analysis, developed here, is applicable to all the digital loops of this class, irrespective of the characteristic of the quantizer device.

SYSTEM MODEL AND ANALYSIS

A block diagram of the first-order DPLL to be discussed is depicted in Fig.1. The input signal, denoted by:

$$x(t) = A \sin[(\omega_0 + \Delta\omega)t + \vartheta_0], \tag{1}$$

is sampled at time instants determined by a digital clock, whose nominal period is $T_0 \triangleq 2\pi/\omega_0$. This sampled output, after being quantized by $Q(\cdot)$ and filtered (in a first-order DPLL the filter is modeled as an amplifier with gain $\mathbf{g}$) is used to adjust the sampling instants in such a way that the clock tracks the positive going zero crossings of the incoming signal.

Remarking the functional dependence on the integer k, instead of the sampling instant t(k), the sampled output, as deduced in (2), is:

$$X(k) = A \sin \phi(k), \tag{2}$$

$\phi(k)$ being the phase error between the input signal and the clock reference. Again in (2) the basic difference equation of DPLL operation is found to be:

$$\phi(k+1) = \phi(k) - \lambda_1 Q[X(k)] + \lambda_2, \tag{3}$$

with the initial condition:

$$\phi(0) \triangleq \phi_0 = \vartheta_0 .$$

In (3) λ_1 and λ_2 are defined as:

$$\lambda_1 \triangleq (\Delta\omega/\omega_0 + 1)\lambda , \tag{4a}$$

$$\lambda_2 \triangleq 2\pi\Delta\omega/\omega_0 , \tag{4b}$$

where $\lambda = \mathbf{g}\omega_0$ is referred as the model gain. Furthermore the time required for the phase error to reach a value $\phi(k)$ is expressed by:

$$t(k) = \frac{\phi(k) - \phi_0 + 2k\pi}{\omega_0 + \Delta\omega} . \tag{5}$$

It is worthwhile remarking that, since $t(k+1) \geqslant t(k)$, the model gain must be [see (1),(2) and (3)] such that $\lambda Q_M \leqslant 2\pi$, Q_M being the maximum absolute value of $Q(\cdot)$. Note that such an upper bound on the loop gain is not present for an analog PLL, where any loop gain can be used.

In the following it will be evident that the analysis here presented is applicable to all digital loops of this class, irrespective of the characteristic of the quantizer device. Therefore, for the sake of clarity and brevity, we limit ourselves to consider only a type of dead zone quantizer, which is specified by its maximum number of positive levels, say L/2, and by the values of the input signals at which the level changes occur. Since the input signal is of the type $i = I \sin\Phi$, we suggest that the values of level changing points be chosen according to the rule:

$$i_n = \pm I \sin \frac{2n-1}{2}\delta \quad , (n = 1, 2, \ldots)$$

δ being π/L. In this way the output characteristic of the device results in a uniform quantization with respect to the argument, as shown in Fig.2, where the size of the steps is normalized to the unity and an odd characteristic is chosen for $Q(\cdot)$.

We now shortly outline the graphical method which allows one to fully investigate the acquisition behavior of the loop. Let us consider the following functions of the continuous variable ϕ:

$$c(\phi) = \phi - \lambda_1 Q(A \sin\phi) \tag{6a}$$

$$d(\phi) = \phi - \lambda_2 \tag{6b}$$

referred to as the characteristic curve and the detuning straight line respectively. With reference to the quantizer previously defined and looking at Fig.3, where the case L=6 is shown, we project horizontally the point $P_0[\phi_0,\ c(\phi_0)]$ on the point $P'_0[c(\phi_0)+\lambda_2,\ c(\phi_0)]$ of the detuning line, then P'_0 vertically on the point P_1 of the characteristic curve and so on. Since $d[\phi(k+1)] = c[\phi(k)]$ the sequence of the abscissae of points P_k coincides with the sequence $\{\phi(k)\}$ of the phase error at each sampling. Moreover the number of horizontal segments of such trajectory represents the number of samplings required to reach the phase error $\phi(k)$ since the r-th horizontal segment $P_rP'_r$ represents the phase correction produced by the r-th sampling. By utilizing this graphical method, it is easy to show that, if $|\lambda_2|>\lambda_1 L/2$, the phase error always increases or decreases according to whether $\lambda_2 \gtrless 0$. However, an only necessary condition to achieve the synchronization is $|\lambda_2|\leq\lambda_1 L/2$, i.e.:

$$-\frac{\lambda L/2}{2\pi + \lambda L/2} \leq \frac{\Delta\omega}{\omega_0} \leq \frac{\lambda L/2}{2\pi - \lambda L/2} \tag{7}$$

In the particular case when $|\lambda_2|=m\lambda_1$ $(m=0,1,\dots,L/2)$, the points which belong to both $c(\phi)$ and $d(\phi)$ represent the possible stationary solutions of eq.(3). Such solutions are neutral in the sense that any perturbation will shift them in proportion to the perturbation magnitude. In general perfect locking cannot be achieved since it is not possible, by discrete corrections, to bring the clock in perfect alignment with the incoming signal. Nevertheless, by a suitable choice of the loop parameters, the final phase error can be restricted in a prefixed range, as shown in the next sections.

PHASE STEP INPUT

For a phase step input $\Delta\omega=0$, i.e. $\lambda_1=\lambda$ and $\lambda_2=0$, so that the characteristic curve and the detuning line result as in Fig.4. With reference to the same figure, let us observe that, if at j-th sampling $\phi_M < \phi(j) < \phi_N$, at least one cycle is skipped. We notice that, depending on the initial conditions, the model gain and the number of the quantization levels, the number of cycles skipped can be great, even infinite. In this case the loop phase error may cycle indefinitely through multiples of 2π radians so that synchronization is never achieved. To avoid any cycle slipping, it is sufficient that the ordinate $(\pi+\delta/2)$ of point B in Fig.4 be greater than the ordinate $(\lambda L/2-\pi/2+\delta/2)$ of point A, i.e. the loop parameters must be chosen according to the rule $L\lambda < 3\pi$ or, recalling that $L\delta=\pi$,:

$$\lambda < 3\delta \tag{8}$$

For the loop design it is of great interest to select the loop parameters so that the final phase error be restricted in a prefixed range, say $(-\varepsilon,\ \varepsilon)$. Since the derivation of this condition is too laborious to be included here, we limit ourselves to maintain that, if:

$$\delta\,\frac{2m-1}{m} < \lambda < \delta\,\frac{2m+1}{m+1}\ ,\quad (m = 0,\ 1,\dots,L/2) \tag{9a}$$

the result is:

$$\varepsilon = m(\lambda-\delta) + \delta/2\ . \tag{9b}$$

Obviously the condition (9a) is more stringent than the condition (8) and, consequently, it excludes any cycle slipping. Equations (9) entail that only the following solutions are possible (see Fig.5):

i) oscillating solutions of period two whose trajectories touch at most the levels ±m ($m \leq L/2$) belonging to the characteristic curve $c(\phi)$ in the interval $[-(\pi+\delta)/2,\ (\pi+\delta)/2]$ (see e.g. the dotted regions in Fig.5 where m=2);

ii) stationary solutions in the range $(-\delta/2,\ \delta/2)$;

iii) stationary solutions in the ranges $[-(\pi+\delta/2),\ -(\pi-\delta/2)]$ and $[\pi-\delta/2,\ \pi+\delta/2]$, corresponding to the dead zones around $\pm\pi$ where the negative going zero crossings of the incoming signal are tracked.

Since from (9a) we have $\lambda<2\delta$, by geometrical arguments it is easy to show that the dead zones around $\pm\pi$ are entered only if the initial phase error is such that $\left||\phi_0| - \pi\right| < \delta/2$. Let us note that, if a perturbation pushes the phase error out of the aforesaid ranges, a next dead zone is reached, where the positive going zero crossings of the signal are tracked.

Moreover, we point out that the limit cycles i) are neutral since the slope of each climbing segment of $c(\phi)$ is equal to the unity.

At this point we recall that the basic purposes of employing a dead zone quantizer is to avoid corrections to the clock period when the magnitude of the phase error becomes less than the dead zone extension $\delta/2$. To take full advantage of this fact, we must avoid any oscillating solution. This is accomplished if the loop gain and the dead zone extension are chosen in such a way that eq. (9a) holds true with m=0, i.e.:

$$\lambda \leqslant \delta \tag{10a}$$

Consequently the final phase error range (9b) becomes:

$$\varepsilon = \delta/2\ . \tag{10b}$$

PHASE ACQUISITION TIME

When the loop is intended for a phase step acquisition, at first we must settle the final phase error range ε. Then, to avoid the oscillating solution, by eqs. (10) we determine δ and the upper-bound of the loop gain λ. We note that the number of the quantization levels is $L = \pi/\delta$.

If the choice $\lambda = \delta$ is made, from Fig.6 it is easy to deduce that only one sampling is necessary to reach the stationary solution when the initial phase error is less than $(\pi+\delta)/2$ in magnitude. The same graphical approach shows that this value of the loop gain assures the faster acquisition for all values of the initial phase error, even if the angular frequency of the input carrier is deviated by an amount $\Delta\omega \ll \delta/T_0$. Therefore we limit ourselves to evaluate the phase acquisition time in the case $\lambda = \delta$, sketched in Fig.6 where L=12.

When $(\pi+\delta)/2 < \phi_0 < \pi-\delta/2$, as one can visually ascertain by inspecting Fig.6, the phase error is corrected at the first sampling by an amount:

$$\Delta_1 \triangleq \phi(0) - \phi(1) = (L - n)\delta \tag{11}$$

n being the integer such that:

$$(2n - 1)\delta/2 < \phi_0 < (2n + 1)\delta/2\ .$$

This entails:

$$n = (\phi_0/\delta - 1/2)_{u.i.} \tag{12}$$

where the subscript u.i. denotes the rounding off of the number

$(\cdot)$ to the upper integer. The same figure places in evidence the fact that the value of each correction, say Δ_j, is:

$$\Delta_j \triangleq \phi(j-1) - \phi(j) = 2^{j-1}\Delta_1, \tag{13}$$

as long as $\phi(j-1) > (\pi+\delta)/2$. Therefore the minimum number μ of samplings necessary for the phase to become less than $(\pi+\delta)/2$ is implicitly expressed by:

$$\sum_{j=1}^{\mu} \Delta_j = \Delta_1 (2^{\mu} - 1) \geqslant \phi_o - \frac{\pi+\delta}{2} . \tag{14}$$

In conclusion, bearing in mind that only one sampling is necessary to reach a stationary value immediately after $\phi(\mu)$ became less than $(\pi+\delta)/2$ and collecting (11), (12), (13) and (14), the overall number of samplings for the acquisition is:

$$s = \mu + 1 = \left. \frac{\ln \dfrac{L - 1 - 2\left[\left(\frac{L|\phi_o|}{\pi} - \frac{1}{2}\right)_{u.i.} - \frac{L|\phi_o|}{\pi}\right]}{L - \left(\frac{L|\phi_o|}{\pi} - \frac{1}{2}\right)_{u.i.}}}{\ln 2} \right|_{u.i.} \tag{15}$$

where use has been made of eq. $L\delta = \pi$ and of the absolute value of ϕ_o, since s is independent of the sign of the initial phase error.

The phase acquisition time T_p can be obtained by evaluating eq. (5) for k=s and by looking at Fig.6 from which it results $\phi_o - \phi(s) = n\pi/L$, where n is again expressed by eq. (12). Therefore we have:

$$T_p \triangleq t(s) = \left(2s - \frac{n}{L}\right)\frac{\pi}{\omega_o} . \tag{16}$$

In order to place in evidence the impact of the number L of the quantization levels on the loop performances, a reasonably straightforward expedient is to evaluate (5) and (15) when the initial phase error assumes the maximum value $|\phi_M| \not< \pi - \delta/2$ for which the loop still succeeds in positive going zero crossing synchronizing. We note that, for a given L, the largest number s_{max} of the samplings for the acquisition and, consequently, the largest phase acquisition time $T_{p\ max}$ occur when $\phi_o = \phi_M$. Under this condition we obtain:

$$s_{max} = \left(\frac{\ln L}{\ln 2}\right)_{u.i.} \tag{17a}$$

$$T_{p\ max} \cong (2s_{max} - 1)\pi/\omega_o \tag{17b}$$

In conclusion $T_{p\ max}$ results in a monotonic increasing function of L ($L \geqslant 2$) so that, to obtain the faster acquisition, we must settle:

$$L = 2(\pi/4\varepsilon)_{u.i.} ,$$

ε being chosen according to the final phase error permitted by the specific application. The trade-off between ε and $T_{p\ max}$ is quite evident.

FREQUENCY STEP INPUT

In order to characterize meaningfully the acquisition behavior of the loop, we must evaluate the pull-in range. This is defined as the total offset of the angular frequency around ω_o, inside which any input frequency will result in locking, i.e. the final phase error will result "suitably" bounded after a certain transient.

Let us note that inequalities (7) do not determine the pull-

-in range, as is suggested in literature. Infact, looking at Fig.3, we see that, if $\phi_V < \phi(j) < \phi_W$, at least one cycle is skipped and, what is more, many or even infinite cycles can be skipped depending on the initial phase error, the initial detuning and the loop gain. A sufficient condition to avoid this drawback can be specified as follows: "the ordinate of point B in Fig.3 must be greater than the ordinate of point A". Bearing in mind that, to have synchronization, we require at most:

$$\lambda_2 = \lambda_1 L/2 \tag{18}$$

(in this case the point B shifts upon B' in Fig.3), the previous condition entails:

$$\lambda_1 \leq \delta \tag{19}$$

By collecting this inequality with eqs. (4) and (18) we find first an upper-bound to the loop gain:

$$\lambda \leq 4\delta/5 \tag{20}$$

and then from (7) we obtain the largest normalized frequency offset:

$$-\frac{1}{6} < \frac{\Delta\omega}{\omega_0} < \frac{1}{4} \tag{21}$$

for which this class of loops will lock regardless of the initial conditions and without any cycle slipping.

Under assumption (20) and with reference to Fig.7, where for the sake of simplicity only a few levels and a positive detuning are considered, it is easy to show that only the following cases are possible:

i) stationary solutions exist in two phase ranges whose extent is:

$$2\varepsilon = \delta \tag{22a}$$

around the abscissae $m\delta\ \mathrm{sgn}(\lambda_2)$ and $(\pi - m\delta)\,\mathrm{sgn}(\lambda_2)$, if $|\lambda_2| = m\lambda_1$;

ii) after a certain sampling the phase error fluctuates in a range whose extent is:

$$2\varepsilon = \lambda_1 \tag{22b}$$

around the abscissa:

$$\phi_1 = \frac{2m - 1}{2}(\delta - \lambda_1)\,\mathrm{sgn}(\lambda_2) + \lambda_2$$

where m is implicitly defined by:

$$(m - 1)\lambda_1 < |\lambda_2| < m\lambda_1 ;$$

this is a result of the fact that, beginning from a certain sampling, all possible trjectories, as A B C . . . in Fig.7, touch only (m-1)-th and m-th levels.

We observe that the stationary solutions in the range around $(\pi - m\delta)\ \mathrm{sgn}(\lambda_2)$ take place only if ϕ_0 belongs to the same range. Moreover, if a perturbation pushes the phase error out of the aforesaid range, a nearest stationary solution range is reached. As far as case ii) is concerned, we maintain that a periodic solution with period p exists if and only if an integer q<p exists such that:

$$q(m\lambda_1 - |\lambda_2|) = (p-q)\left[|\lambda_2| - (m-1)\lambda_1\right] .$$

This condition can be explained by looking at Fig.7, where the case p=5 and q=3 is illustrated. Let us observe that, to have for every p samplings the closure of the trajectory, the summ of the q corrections towards left (each of magnitude $m\lambda_1 - |\lambda_2|$) must be equal to the summ of the (p-q) corrections towards right (each of magnitude $|\lambda_2| - (m-1)\lambda_1$). These limit cycles are neutral, too.

In conclusion, it should be stressed that the largest normalyzed frequency offset (21) which can be tracked by this class of loops, without cycles slipping, is independentof the number L of quantization levels, provided that the loop gain is $\lambda=4\delta/5$. In this case, since δ represents the maximum range of the final phase error fluctuation (see eqs. (19) and (22)), we choose δ according to the specific application and immediately obtain the levels number $L=\pi/\delta$ and the loop gain.

CONCLUSIONS

The acquisition performances of an all digital first-order phase-locked loop with a multilevel quantizer has been considered in a noiseless case. This is accomplished by a graphical method which offers a direct physical insight into the loop behavior for both phase and frequency step inputs. The analysis places in evidence what bounds must be imposed on the loop gain in order either to avoid cycle slippings or to obtain that the final phase error fluctuates in a suitably small range. Moreover, the limit cycles phenomemon has been fully investigated with regards to the existence, periodicity and stability. Finally a closed-form expression of phase acquisition time has been drawn for a loop devised for the faster attainment of a stationary final phase error.

References

(1) - S.C. GUPTA: "Status of Digital Phase Locked Loops", Proc. IEEE, vol. 63, N°2, pp. 291-306, Feb. 1975.

(2) - C.P. REDDY, S.C. GUPTA: "A Class of All Digital Phase Locked Loops: Modeling and Analysis", IEEE Trans. Ind. Electron. Contr. Instrum., vol. IECI-20, pp. 239-251, Nov. 1973.

(3) - F. RUSSO: "Anelli Numerici ad Aggancio di Fase", Internal Report of 'Centro Studi per i Metodi ed i Dispositivi di Radiotrasmissione' of CNR, ed. Tecnico Scientifica, Dec. 1976, PISA.

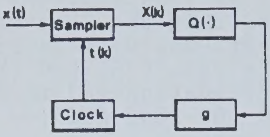

x(t)
Sampler
X(k)
Q(·)
t(k)
Clock
g

Fig.1. Block diagram of DPLL

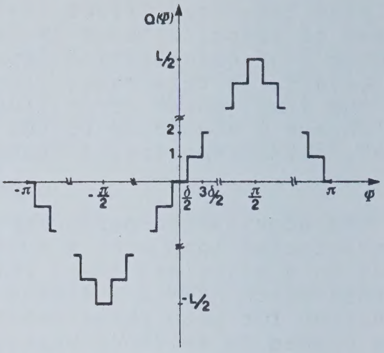


Fig.2. Multilevel quantizer with dead zone

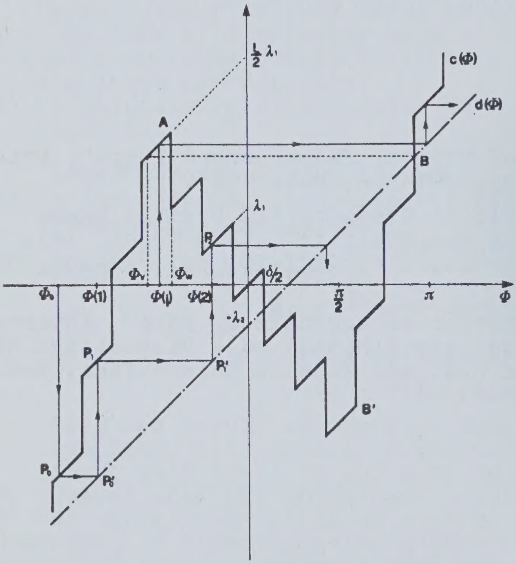

L/2 λ1
c(Φ)
d(Φ)
A
B
λ1
P
Φv
Φw
δ/2
Φ0
Φ(1)
Φ(j)
Φ(2)
π/2
π
Φ
-λ2
P1
P1'
B'
P0
P0'

Fig.3. Description of the graphical method

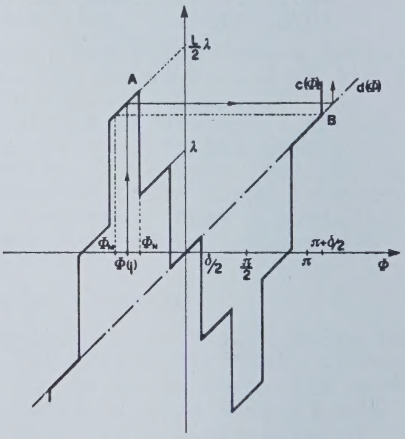

L/2 λ
A
c(Φ)
d(Φ)
B
λ
Φw
Φw
Φ(j)
π+δ/2
δ/2
π/2
π
Φ

Fig.4. Cycles slipping phenomenon for a phase step input

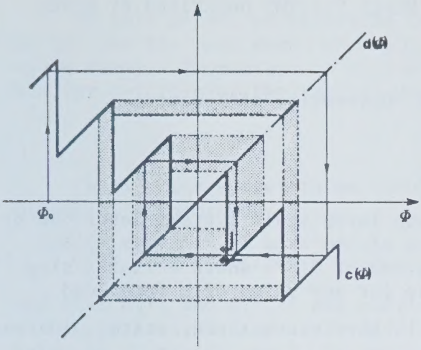


Fig.5. Example of regions where limit cycles take place

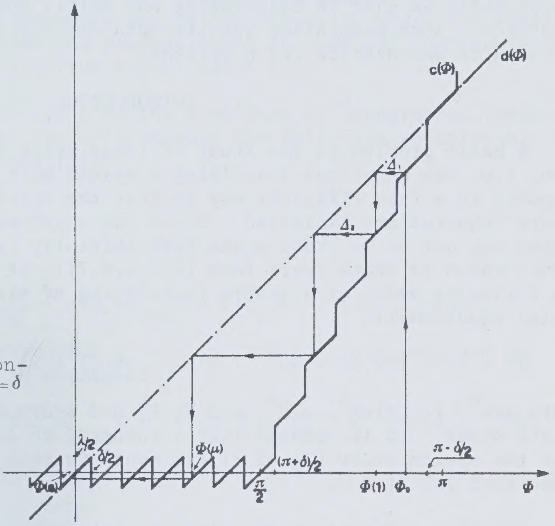

d(Φ)
c(Φ)
Δ₁
Δ₂
λ/2
δ/2
Φ(μ)
(π+δ)/2
π - δ/2
Φ(0)
π/2
Φ(1)
Φ₀
π
Φ

Fig.6. The graphical construction when $\lambda = \delta$

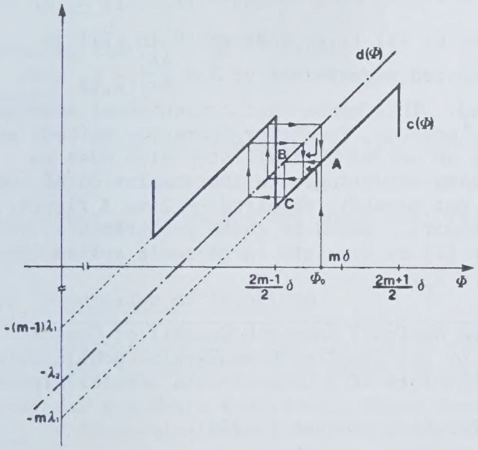

d(Φ)
c(Φ)
B
A
C
mδ
(2m-1)/2 δ
Φ₀
(2m+1)/2 δ
Φ
-(m-1)λ₁
-λ₂
-mλ₁

Fig.7. Limit cycles phenomenon for a frequency step input

AN ALGORITHM FOR THE SIMULATION OF LARGE SCALE SYSTEMS DESCRIBED BY MIXED ALGEBRAIC AND DIFFERENTIAL EQUATIONS

N.K. TRIPATHI* and E.J. DAVISON
Department of Electrical Engineering, University of Toronto,
Toronto, Ontario, Canada M5S 1A4

ABSTRACT

An algorithm is described to integrate large scale systems described by mixed algebraic and differential equations of the type $T\dot{x} = Ax+f(x,t)+Bu$. The proposed algorithm has a truncation error of $0(h^3)$ where h is the step size of integration, is numerically stable for any value of h (provided $T\dot{x} = \left(Ax + \left.\frac{\partial f}{\partial x}\right|_{(\bar{x},\bar{t})}\right) x$ is stable), and will give exact steady state solutions when u is constant for any value of h. It is shown that sparse techniques can directly be used in calculating any matrix operations required by the algorithm. Some simulation results obtained for a 98th order systems are included to demonstrate the algorithm.

1. INTRODUCTION

A basic problem in the study of large scale systems is that of simulation, i.e. the equations describing a system must be integrated on a digital computer in a fast efficient way so that the total computation time and the memory required are minimized. Since the equations describing a physical system may not be in state space form initially (and it may not be easy to reduce these to state space form [2,3,4,6,7]), it is desirable to consider the following model of a system (consisting of mixed algebraic and differential equations):

$$T\dot{x} = Ax + f(x,t) + Bu \qquad x(0) = x_0 \tag{1}$$

where $x \epsilon R^n$, $f(x,t) \epsilon R^n$, $u \epsilon R^m$, and T, A, and B are constant matrices of appropriate order. It is assumed that a solution to (1) exists, is unique, and that the system order [7] of (1) is equal to rank T; in addition, it is assumed that $|sT-A| \not\equiv 0$.

The class of systems described by (1) includes population growth models, chemical processes, boiler systems, power systems and electric networks etc.

A large scale system described by (1) (i.e. when $n >> 10$ in (1)) is usually stiff [5], i.e. the generalized eigenvalues of $A + \left.\frac{\partial f}{\partial x}\right|_{(\bar{x},\bar{t})}$ with respect to T are widely distributed. This means that conventional methods of integration such as Runga-Kutta methods, Predictor Corrector methods etc. cannot be used to solve (1), since an extremely small step size must be used to prevent numerical instability occurring. In the simulation of such systems however, high accuracy is not normally required — 2 to 3 figure accuracy often being more than adequate. Based on these requirements, an algorithm was developed by Davison [1] to simulate large scale system described by (1) for the case when $T = I_n$.

This work has been supported by the National Research Council of Canada under Grant No. 4396, and in part by the Canadian Commonwealth Scholarship and Fellowship Administration in the form of a Commonwealth Scholarship to the first author.
*On leave from the University of Roorkee, Roorkee (India).

In this paper, an algorithm is developed to simulate the system described by (1) for the case when rank $T \leq n$ (i.e. it is not necessary to reduce (1) to state space form). Properties of this algorithm are studied, and a numerical method for determining the initial conditions for (1) is given.

2. DEVELOPMENT

In (1), the system to be simulated, it is assumed that $f(x,t)$ satisfies Lipschitz's condition:

$$|f(x_A,t) - f(x_B,t)| \leq K_I|x_A - x_B|, \; \forall t \in [0, \infty), \; \forall x_A, x_B \in R^n \tag{2}$$

and that $u(\cdot)$ and $f(\cdot,\cdot)$ are sufficiently smooth so that $x(t)$ is differentiable at least 3 times. In addition it is assumed that n is generally large (e.g. $n >> 10$) and K_I is small. It should be noted that systems which satisfy these properties occur often in engineering and physics, e.g. electrical and mechanical systems where saturation occurs or where the nonlinearities are of the sine, cosine and product type; in chemical engineering where nonlinear vapor liquid equilibrium curves exist etc.

Let x_t denote $x(t)$ and let h be the time step of integration (step size). By expanding x_{t+h} and x_t by Taylor's formula the following relationship between x_{t+h} and x_t is obtained [1]:

$$x_{t+h} = x_t + \frac{h}{2}(\dot{x}_t + \dot{x}_{t+h}) \tag{3}$$

and from (1):

$$T(\dot{x}_t + \dot{x}_{t+h}) = A(x_t + x_{t+h}) + B(u_t + u_{t+h}) + f(x_t,t) + f(x_{t+h},t+h) \tag{4}$$

On substituting the expression for $\dot{x}_t + \dot{x}_{t+h}$ from (3) into (4), and on simplifying, the following is obtained:

$$x_{t+h} = S^- S^+ x_t + S^- \frac{h}{2} B(u_t + u_{t+h}) + S^- \frac{h}{2} f(x_t,t) + S^- \frac{h}{2} f(x_{t+h},t+h) + O(h^3) \tag{5}$$

where

$$S^- \triangleq (T - \frac{h}{2}A)^{-1} \text{ and } S^+ = (T + \frac{h}{2}A)$$

assuming that $(T - \frac{h}{2}A)^{-1}$ exists.

Remark

Note that since it has been assumed that $|sT-A| \not\equiv 0$, then the matrix $(T - \frac{h}{2}A)^{-1}$ exists for "almost all" values of h, i.e. $(T - \frac{h}{2}A)^{-1}$ exists for all values of h, except for those real values of h which lie on the hypersurface $|T - \frac{h}{2}A| = 0$.

2.1 Discussion of Algorithm

Equation (5) forms the basis of a recursive algorithm for calculating x_{t+h} given x_t, where x_{t+h} is given implicitly as the solution of a set of nonlinear algebraic equations. These equations can rapidly be solved for x_{t+h} by successive approximation if the Lipschitz's constant K_I of (1) is small [1]. The only data needed for solving (5) is A, T, B, u(•) and f(•,•) which is normally available for the class of systems under considerations. x(0) may or may not be available and can be obtained as described in section 3. S^-S^+ and S^- are the only n^3 operations required in the algorithm, and if h is kept constant, S^-S^+, S^- need be computed only once during a simulation run. It should be noted that if T and A are sparse, then the matrix S^- can easily be calculated by using sparse matrix techniques for matrix inversion; the same comment applies for the calculation of S^-S^+.

2.2 Properties of the Algorithm

Definition

The system $T\dot{x} = Ax$ is asymptotically stable if and only if $\lambda_i \in \mathbb{C}^-$, $i = 1,2,\ldots,n$ where $\mathbb{C}^-$ is the open left half of the complex plane and λ_i, $i = 1,2,\ldots,n$ are the generalized eigenvalues of A with respect to T [10].

The following generic result is obtained from [9].

Lemma 1 [9]

Given $T\dot{x} = Ax + Bu$ assume that T is fixed and that $|sT-A| \not\equiv 0$; then for "almost all" A the system $T\dot{x} = Ax + Bu$ has system order equal to rank T, i.e. the class of matrices A in which this is not true lies on a hypersurface in the parameter space of A.

Properties of Algorithm

I The algorithm given by (5) has a truncation error of order $0(h^3)$ where

$$0(h^3) = \frac{h^3}{24}\left(\frac{d^3x}{dt^3}\right) \tag{6}$$

This follows immediately from the development of (5).

II The algorithm is numerically stable for the system $T\dot{x} = \left\{A + \left.\frac{\partial f}{\partial x}\right|_{(\bar{x},\bar{t})}\right\}x$ for any value of h>0, provided that the system $T\dot{x} = \left\{A + \left.\frac{\partial f}{\partial x}\right|_{(\bar{x},\bar{t})}\right\}x$ is asymptotically stable and has system order equal to rank T.

Proof

Let $H \triangleq \left\{T - \frac{h}{2}\left(A + \left.\frac{\partial f}{\partial x}\right|_{(\bar{x},\bar{t})}\right)\right\}^{-1}\left\{T + \frac{h}{2}\left(A + \left.\frac{\partial f}{\partial x}\right|_{(\bar{x},\bar{t})}\right)\right\}$; then algorithm (5) is numerically stable for the system $T\dot{x} = \left(A + \left.\frac{\partial f}{\partial x}\right|_{(\bar{x},\bar{t})}\right)x$ if there exists a constant c>0 so that $\lim_{k\to\infty}\|H^k\| < c$.

Now choose $P \in R^{n\times n}$, $Q \in R^{n\times n}$ where P, Q are nonsingular so that PTQ and $P\left(A + \left.\frac{\partial f}{\partial x}\right|_{(\bar{x},\bar{t})}\right)Q$ are in upper triangular form [10]. Then

$$Q^{-1} H Q = \hat{R}\hat{S} \tag{7}$$

where $\hat{R}$, $\hat{S}$ are upper triangular matrices given by:

$$\hat{R} \triangleq \left\{P\left(T - \frac{h}{2}\left(A + \left.\frac{\partial f}{\partial x}\right|_{(\bar{x},\bar{t})}\right)\right)Q\right\}^{-1}$$

$$\hat{S} \triangleq P\left[T + \frac{h}{2}\left(A + \frac{\partial f}{\partial x}\bigg|_{(\bar{x},\bar{t})}\right)\right]Q$$

Let the diagonal elements of $\hat{R}$, $\hat{S}$ now be denoted by $(t_i - \frac{h}{2} a_i)^{-1}$ and $(t_i + \frac{h}{2} a_i)$, $i = 1,2,..,n$ respectively and let rank $T \triangleq n^*$; then from (7)

$$H = PQ\begin{pmatrix} \lambda & 0 \\ 0 & -I_{n-n^*} \end{pmatrix}Q^{-1}P^{-1}$$

where P is a suitable permutation matrix, and λ is an upper triangular matrix which has diagonal elements given by $\xi_i \triangleq \dfrac{\bar{t}_i + \frac{h}{2}\bar{a}_i}{\bar{t}_i - \frac{h}{2}\bar{a}_i}$, $i = 1,2,\ldots,n^*$. Now if the system $T\dot{x} = \left(A + \frac{\partial f}{\partial x}\Big|_{(\bar{x},\bar{t})}\right)x$ is stable, this implies that $|\xi_i| < 1$, $i = 1,2,\ldots,n^*$ and hence it is concluded that there exists a constant $c>0$ so that $\lim_{k\to\infty} \| H^k\| = \| PQ\begin{pmatrix} \lambda^k & 0 \\ 0 & (-I_{n-n^*})^k \end{pmatrix}Q^{-1}P^{-1}\| < c$.

III The following result is now obtained regarding the steady-state behaviour of algorithm (5).

Theorem 1

Consider the system (1) for the case of constant input u_{ss} and assume that (1) is uniformly asymptotically stable in the large, in the sense of Lyapunov, about the equilibrium point x_{ss} where x_{ss} satisfies the equation $Ax_{ss} + Bu_{ss} + \bar{f}(x_{ss}) = 0$ where $\bar{f}(x_{ss}) \triangleq \lim_{t\to\infty} f(x_{ss},t)$. Let an input $u(t)$ be applied to (1) so that $\lim_{t\to\infty} u(t) = u_{ss}$, a constant vector; then if the algorithm (5) converges to a steady state value as $t\to\infty$, it converges to the exact steady state solution of (1).

Proof

Suppose that as $t\to\infty$

$$x_{t+h} = x_t = x_{ss}; \quad f(x_{t+h},t+h) = f(x_t,t) = \bar{f}(x_{ss})$$

then from (5)

$$x_{ss} = S^- S^+ x_{ss} + S^- hBu_{ss} + S^- h\bar{f}(x_{ss})$$

and on substituting for S^- and S^+, the following is obtained:

$$\{I - (T - \frac{h}{2} A)^{-1}(T + \frac{h}{2} A)\}x_{ss} = (T - \frac{h}{2} A)^{-1}h\{Bu_{ss} + \bar{f}(x_{ss})\}$$

which implies that

$$-Ax_{ss} = Bu_{ss} + \bar{f}(x_{ss}) \tag{9}$$

and thus that x_{ss} is the exact steady state solution of (1).

This shows that the algorithm gives correct steady state solutions for (1) for any value of h (and not just for $h\to 0$). This property is important because it (together with a truncation error of $0(h^3)$) implies that the solution of (1) will be relatively accurate for large values of h.

IV The algorithm has the property that the nonzero elements of $T-\frac{h}{2}A$, $T+\frac{h}{2}A$ correspond to the nonzero elements of T or A. This property is very useful in large scale systems when T and A are sparse (in which case, sparse matrix technqiues [11] can then be used in the algorithm).

V The nonlinear equation (5) may be solved for x_{t+h} given x_t by the method of successive approximations, if the Lipschitz constant of $f(\cdot,\cdot)$ is sufficiently small, in particular, if $K_I < \frac{2}{h\|S^-\|}$ [1].

Remark on System Order of (1)

It should be noted, that if $f(x,t) \equiv A^*x$ in (1), then the system order of (1) is equal to rank T for "almost all" A, A* matrices from lemma 1. Thus, this means that the assumption that (1) has system order equal to rank T is not unduly restrictive.

3. DETERMINATION OF THE INITIAL CONDITION FOR (1)

For systems described in state space form, the initial condition is arbitrary, but for systems described by (1), this is not necessarily the case. An algorithm is now described to obtain an intial condition which is consistant with (1).

Let P, Q be nonsingular matrices with the property that $PTQ = \begin{pmatrix} T_1 & 0 \\ 0 & 0 \end{pmatrix}$ where T_1 is nonsingular, with rank T_1 = rank $T \triangleq n^*$, and let

$$x = Qz \tag{10}$$

Then (1) becomes, on multiplying by P, the following:

$$PTQ\dot{z} = PAQz + PBu + Pf(Qz,t) \tag{11}$$

which may be written as:

$$\begin{pmatrix} T_1 & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \end{pmatrix} = \begin{pmatrix} A_1 & A_2 \\ A_3 & A_4 \end{pmatrix}\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} + \begin{pmatrix} B_1 \\ B_2 \end{pmatrix}u + \begin{pmatrix} f_1(Qz,t) \\ f_2(Qz,t) \end{pmatrix} \tag{12}$$

where $\begin{pmatrix} A_1 & A_2 \\ A_3 & A_4 \end{pmatrix} \triangleq PAQ$, $\begin{pmatrix} B_1 \\ B_2 \end{pmatrix} \triangleq PB$, $\begin{pmatrix} f_1 \\ f_2 \end{pmatrix} \triangleq Pf$

Choose now $z_1(0)\epsilon R^{n^*}$ arbitrary and solve the following equation for $z_2(0)$:

$$A_4z_2(0) = -A_3z_1(0) - B_2u(0) - f_2\left[Q\begin{pmatrix} z_1(0) \\ z_2(0) \end{pmatrix},0\right] \tag{13}$$

An unique solution for $z_2(0)$ exists since, by assumption, the system order of (1) is equal to rank $T \triangleq n^*$, and by assumption, (1) possesses an unique solution. Thus the initial condition for (1) is given by:

$$x(0) = Q\begin{pmatrix} z_1(0) \\ z_2(0) \end{pmatrix} \tag{14}$$

where $z_1(0)\epsilon R^{n^*}$ is arbitrary, and where $z_2(0)$ is given by the solution of the nonlinear algebraic equation (13). It may be noted that if the Lipschitz constant of $f(\cdot,\cdot)$ is "small enough", and if the system order of (1) is equal to rank T then A_4 will be nonsingular, and (13) may be solved for $z_2(0)$ by using successive approximations.

4. NUMERICAL EXAMPLE

Example 1 - 98th Order Interconnected Power System

The interconnected power system of [2] with 3 areas and 9 synchronous machines, and described by a 98th order system was simulated using the proposed algorithm. Some results of the simulation for a unit step function input in torque disturbance for machine 6, with all initial conditions zero, are given in Table 1 (with h = 0.4 sec).

Table 1 Simulation Results Obtained for Power System

Time in seconds	Results obtained using the proposed algorithm		
	x_1	x_9	x_{58}
4	0.109×10^{-3}	0.234×10^{-1}	0.107×10^{-1}
8	0.754×10^{-4}	0.241×10^{-1}	0.107×10^{-1}
12	0.840×10^{-4}	0.241×10^{-1}	0.107×10^{-1}
16	0.834×10^{-4}	0.241×10^{-1}	0.107×10^{-1}

The computation time required to simulate this sytem (on an IBM 370/120 digital computer) was approximately 30 seconds. The results reported above are accurate from 1 to 3 significant figures; greater accuracy could be achieved by using a smaller step size.

5. CONCLUSIONS

An algorithm has been proposed for the computer simulation of a large scale system described by mixed algebraic and differential equations. In this algorithm, it is not necessary to reduce the system to state space form. Various properties of the algorithm are described; in particular, it is shown that the algorithm has a truncation error of $O(h^3)$, is numerically stable for any value of the step size h, and gives exact steady-state solutions (for constant inputs) for any h. In addition, it is shown that sparse matrix methods may be used in the proposed algorithm; this makes the algorithm particularly attractive for large scale systems.

REFERENCES

[1] Davison, E.J., "An Algorithm for the Computer Simulation of Very Large Dynamic Systems", Automatica, vol. 9, pp. 665-675, 1973.

[2] Davison, E.J., Tripathi, N.K., "Optimal Decentralized Control of a Large Electric Power System — Load and Frequency Control", IEEE Control and Decision Conference, December 1976, pp. 523-530.

[3] Eklund, K., "Numerical Model Building", Int. J. of Control, vol. 11, No. 6, 1970, pp. 973-985.

[4] Gantmacher, F.R., The Theory Matrices, vol. II, Chelsea Publishing Co., 1959, p. 25.

[5] Gear, C.W., Numerical Initial Value Problem in Ordinary Differential Equations, Prentice-Hall, 1971, p. 209.

[6] Godbout, L.F., Jordon, D., "On State Space Descriptions of Linear Differential Equations", Proc. of 1975 Conference on Decision and Control, December 10-12, 1975, Houston, Texas, pp. 835-840.

[7] Rosenbrock, H.H., State Space and Multivariable Theory, Nelson, 1970.

[8] Rosenbrock, H.H., Pugh, A.C., "Contributions to a Heirarchial Theory of Systems", Int. J. of Control, vol. 19, No. 5, 1974, pp. 845-867.

[9] Tripathi, N.K., Davison, E.J., "Large Scale Interconnected System Modelling in State Sapce Form", National Systems Conference, Coimbatore, India, June 1977.

[10] Peter, G., Wilkinson, J.H., "Ax = λBx and the Generalized Eigenvalue Problem", SIAM Numerical Analysis, vol. 7, No. 4, December 1970, pp. 479-492.

[11] Rose, D.J., Willoughby, R.A., (editors), "Sparse Matrices and Their Applications", Proc. IBM Conference, Plenum Press, Princeton, N.J., September 1970.

EIGENVALUE SENSITIVITY OF AGGREGATED MODELS

N.K. SINHA and Z. ELRAZAZ
Department of Electrical Engineering
McMaster University
Hamilton, Canada

ABSTRACT

Expressions are derived for first- and second-order eigenvalue sensitivities of aggregated models with respect to the parameters of the original high-order system. Analytical examples are included to illustrate the results.

1. INTRODUCTION

The aggregation method proposed by Aoki [1] for obtaining low-order models for high-order systems has been shown to be very general, and it has been shown [2] that the earlier approaches of Davison [3] and Mitra [4], which were called projective reduction methods, were special cases of aggregation. A considerable amount of simplification in the computation of the aggregation matrix has also been introduced [5, 6], and recently it has been shown that certain aggregated reduced-order models are also singularly perturbed [7]. This shows that although the aggregation method, developed originally by Aoki, and the singular perturbation method, developed originally by Kokotovic and Sannuti [8] may appear different, in fact they are related.

In all this work, an important problem is the sensitivity of the reduced-order model to variations in the parameters of the original high-order system, since there is always a certain amount of uncertainty about the values of some parameters. In this paper, following Zein El-Din et al. [9], expressions are derived for the first- and second-order eigenvalue sensitivities of the aggregated model to the parameters of the original system. These expressions turn out to be quite simple and easy to manipulate. Numerical examples are included to illustrate the procedure.

2. AGGREGATED MODELS

Consider a large-scale system defined by the equations

$$\dot{x} = Ax + Bu \qquad (1)$$

$$y = Cx \qquad (2)$$

where $x \in R^n$, $u \in R^p$ and $y \in R^q$, with $q \ll n$.

The problem of reduced-order modelling is to obtain the equations

$$\dot{x}_r = Fx_r + Gu \qquad (3)$$

$$y_r = Hx_r$$

where $x_r \in R^m$ with $m \ll n$, so that y_r is a close approximation to y.

Aoki [1] proposed the relationship

$$x_r = Kx \tag{5}$$

where the mxn matrix K, representing a projection from R^n to R^m, was called the aggregation matrix. Such a matrix was shown to satisfy the following relationships

$$F K = K A \tag{6}$$

$$G = K B \tag{7}$$

$$H K = C \tag{8}$$

and, in general, equation (8) would only be satisfied approximately. A minimum-norm solution is obtained by using the pseudoinverse [2], leading to the following relationships

$$F = K A K^+ \tag{9}$$

$$G = K B \tag{10}$$

$$H = C K^+ \tag{11}$$

where

$$K^+ = K^T(K K^T)^{-1} \tag{12}$$

is the pseudoinverse of K and the superscript T represents transposition. Furthermore, a nontrivial solution for F is obtained only if all its eigenvalues are also the eigenvalues of A.

A straightforward method for determining the aggregation matrix, K, [5, 6] is based on the eigenvectors of A^T. For the sake of simplicity, it will be assumed that the eigenvalues of A are distinct, and these will be denoted by λ_1, λ_2, ..., λ_n. The corresponding eigenvectors will be denoted by v_1, v_2, ..., v_n, and the modal matrix

$$V = [v_1 \quad v_2 \ldots v_n] \tag{13}$$

It may be added that extension to the general case of repeated eigenvalues is fairly straightforward, in terms of the generalized eigenvectors.

The eigenvalues of A^T are the same as those of A, but the eigenvectors will be different. Let these be denoted by w_1, w_2, ..., w_n. It follows that $W^T = V^{-1}$, where

$$W = [w_1 \quad w_2 \ldots w_n] \tag{14}$$

assuming that the eigenvectors have been normalized.

The aggregation matrix can be obtained directly as

$$K = R^{-1} W_r^T \tag{15}$$

where R is an arbitrary mxm nonsingular matrix and

$$W_r = [w_1 \quad w_2 \ldots w_m] \tag{16}$$

corresponding to the eigenvalues λ_1, λ_2, ..., λ_m of the original system, which are retained in the aggregated model.

It follows that the choice of K is not unique due to the arbitrariness of K. Making R the identity matrix gives a special case, denoted by K_I.

We have

$$K_I = W_r^T \tag{17}$$

$$F_I = W_r^T A V_r = \mathrm{diag}\,(\lambda_1, \lambda_2, \ldots, \lambda_m) \tag{18}$$

3. FIRST- AND SECOND-ORDER EIGENVALUE SENSITIVITIES

Since there is always some uncertainty about the values of certain parameters of a system, it is of considerable interest to analyse the sensitivity of the eigenvalues of a square matrix A with respect to its parameters. Recently Zein El-Din et al. [9] have derived expressions for first- and second-order sensitivities of the eigenvalues with respect to parameters ξ and η of the system. These are summarized below:

$$\frac{\partial \lambda_i}{\partial \xi} = w_i^T \frac{\partial A}{\partial \xi} v_i \tag{19}$$

$$\frac{\partial v_i}{\partial \xi} = \sum_{j=1, j\neq i}^{n} p_{ji}\, v_j \tag{20}$$

$$p_{ij} = (\lambda_j - \lambda_i)^{-1}\, w_i^T \frac{\partial A}{\partial \xi} v_j, \quad i \neq j \tag{21}$$

The scalars p_{ii} in equations (20) and (21) are undefined but arbitrary, and it is advantageous to take them as zero. Hence, equation (20) may be written as

$$\frac{\partial V}{\partial \xi} = V\,P \tag{22}$$

where $P = [p_{ij}],\ p_{ii} = 0$ (23)

The second-order eigenvalue sensitivities in terms of two parameters, ξ and η, may be written as

$$\frac{\partial^2 \lambda_i}{\partial \xi \partial \eta} = w_i^T \left(\frac{\partial A}{\partial \xi} V p_i + \frac{\partial A}{\partial \eta} V q_i + \frac{\partial^2 A}{\partial \xi \partial \eta} v_i \right) \tag{24}$$

where p_i is the i^{th} column of P, and q_i is the i^{th} column of the Matrix Q defined by

$$Q = [q_{ij}] = [(\lambda_j - \lambda_i)^{-1}\, w_i^T \frac{\partial A}{\partial \eta} v_j]\ ,\ q_{ii} = 0 \tag{25}$$

4. EIGENVALUE SENSITIVITIES OF THE AGGREGATED MODELS

Equations (19) and (24) give the first- and second-order eigenvalue sensitivities of the original high-order system with respect to its parameters. Since m of these eigenvalues are retained in the aggregated model, it is of interest to determine the sensitivities of the eigenvalues of the model with respect to parameters of the high-order system. These sensitivities can be written immediately as (where the subscript r denotes the reduced-order model)

$$\frac{\partial \lambda_{ir}}{\partial \xi} = w_{ir}^T \frac{\partial F}{\partial \xi} v_{ir} \tag{26}$$

$$\frac{\partial v_{ir}}{\partial \xi} = \sum_{j=1,\ j \neq i}^{m} p_{jir} \, v_{jr} \tag{27}$$

$$P_{ijr} = (\lambda_j - \lambda_i)^{-1} \, w_{ir}^T \frac{\partial F}{\partial \xi} v_{jr}, \quad i \neq j \tag{28}$$

and

$$\frac{\partial^2 \lambda_{ir}}{\partial \xi \partial \eta} = w_{ir}^T \quad \frac{\partial F}{\partial \xi} V_r \, p_{ir} + \frac{\partial F}{\partial \eta} V_r \, q_{ir} + \frac{\partial^2 F}{\partial \xi \partial \eta} v_{ir} \tag{29}$$

In these equations, v_{ir} is the eigenvector of F and w_{ir} are the eigenvector of F^T corresponding to the eigenvalue λ_i.

In order to relate these sensitivities to those of the original system, we must determine $\partial F/\partial \xi$ in terms of $\partial A/\partial \xi$. First, we note that

$$F = R^{-1} \, W_r^T \, A \, V_r \, R \tag{30}$$

and since the arbitrary matrix R does not depend upond the parameter ξ, we may write

$$\frac{\partial F}{\partial \xi} = R^{-1} \left(W_r^T A \frac{\partial V_r}{\partial \xi} + W_r^T \frac{\partial A}{\partial \xi} V_r + \frac{\partial W_r^T}{\partial \xi} A \, V_r \right) R \tag{31}$$

$$= \begin{bmatrix} w_1^T \\ w_2^T \\ \vdots \\ w_m^T \end{bmatrix} A \left(\frac{\partial v_1}{\partial \xi} \begin{bmatrix} \frac{\partial v_2}{\partial \xi} & \cdots & \frac{\partial v_m}{\partial \xi} \end{bmatrix} + \frac{\partial A}{\partial \xi} [v_1 \quad v_2 \cdots v_m] \right)$$

$$+ \begin{bmatrix} \frac{\partial w_1^T}{\partial \xi} \\ \frac{\partial w_2^T}{\partial \xi} \\ \vdots \\ \frac{\partial w_m^T}{\partial \xi} \end{bmatrix} A \, [v_1 \quad v_2 \cdots v_m] \tag{32}$$

Substituting the expressions for eigenvector sensitivities from equation (20) and recalling that

$$w_i^T v_j = \delta_{ij} \tag{33}$$

where δ_{ij} is the Kronecker delta, equation (32) simplifies to

$$\frac{\partial F}{\partial \xi} = \text{diag}\ [w_i^T \frac{\partial A}{\partial \xi} v_i],\ i = 1, 2, \ldots, m \tag{34}$$

From equation (19), each element of this diagonal matrix is recognized as the eigenvalue sensitivity of the high-order system to the parameter ξ, for the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_m$ which are retained in the aggregated model. Hence,

$$\frac{\partial F}{\partial \xi} = \text{diag}\ \frac{\partial \lambda_i}{\partial \xi},\ i = 1, 2, \ldots, m \tag{35}$$

Substituting this expression into equation (26) we have

$$\frac{\partial \lambda_{ir}}{\partial \xi} = w_{ir}^T\ \text{diag}\ \frac{\partial \lambda_i}{\partial \xi}\ v_{ir} \tag{36}$$

Equation (36) is a very simple relationship between the eigenvalue sensitivities of the reduced-order model and the original system. It is evident that since the eigenvalue sensitivity of the reduced-order model depends on the eigenvectors of the aggregated model, as well as the eigenvalue sensitivities of the original system, a suitable choice of the matrix R for the aggregation matrix will have some effect on these sensitivities. In the special case when R = I, the two sets of sensitivities will be identical, since w_{ir} and v_{ir} are unit vectors.

In a similar manner, it can be shown that

$$\frac{\partial^2 F}{\partial \xi^2} = \text{diag}\ \frac{\partial^2 \lambda_i}{\partial \xi^2},\ i = 1, 2, \ldots, m \tag{37}$$

and

$$\frac{\partial^2 F}{\partial \xi \partial \eta} = \text{diag}\ \frac{\partial^2 \lambda_i}{\partial \xi \partial \eta},\ i = 1, 2, \ldots, m \tag{38}$$

Substituting equations (35), (37) and (38) into equation (20) the second-order eigenvalue sensitivity of the aggregated model is given by

$$\frac{\partial^2 \lambda_{ir}}{\partial \xi \partial \eta} = w_{ir}^T\ \text{diag}\ \frac{\partial \lambda_i}{\partial \xi}\ V_r\, p_{ir} + \text{diag}\ \frac{\partial \lambda_i}{\partial \eta}\ V_r\, q_{ir} + \text{diag}\ \frac{\partial^2 \lambda_i}{\partial \xi \partial \eta}\ v_{ir} \tag{39}$$

5. NUMERICAL EXAMPLE

To illustrate the results obtained above, a simple example will be used. Consider a single-input single-output third-order system described by the equations

$$\dot{x} = \begin{bmatrix} 0 & 0 & -6 \\ 1 & 0 & -11 \\ 0 & 1 & -6 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u$$

$$y = [0 \quad 1 \quad -2]\, x$$

It is required to obtain the eigenvalue sensitivities of the aggregated second-order model retaining the dominant eigenvalues.

The eigenvalues of the original system are -1, -2, and -3, and the corresponding eigenvector matrix is

$$V = \begin{bmatrix} 3 & -3 & 1 \\ 5/2 & -4 & 3/2 \\ 1/2 & -1 & 1/2 \end{bmatrix}$$

also, the eigenvector matrix for the transpose is

$$W = \begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & -3 \\ 1 & 4 & 9 \end{bmatrix}$$

It may be verified that $W^T V = I$.

Assuming R = I, the aggregation matrix is obtained as

$$K_I = W_r^T = \begin{bmatrix} 1 & -1 & 1 \\ 1 & -2 & 4 \end{bmatrix}$$

and

$$V_r = \begin{bmatrix} 3 & -3 \\ 5/2 & -4 \\ 1/2 & -1 \end{bmatrix}$$

Therefore,

$$F_I = W_r\, A\, V_r = \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix}$$

Also, $w_{1r} = [1 \quad 0]$, $w_{2r} = [0 \quad 1]$, $v_{1r} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $v_{2r} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

Let $\eta = a_{23}$. Hence,

$$\frac{\partial \lambda_1}{\partial \eta} = [1 \quad -1 \quad 1] \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 3 \\ 5/2 \\ 1/2 \end{bmatrix} = -0.5$$

$$\frac{\partial \lambda_2}{\partial \eta} = [1 \quad -2 \quad 4] \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -3 \\ -4 \\ -1 \end{bmatrix} = 2$$

$$\frac{\partial F_I}{\partial \eta} = \begin{bmatrix} -0.5 & 0 \\ 0 & 2 \end{bmatrix}$$

$$\frac{\partial \lambda_{1r}}{\partial \eta} = [1 \quad 0] \begin{bmatrix} -0.5 & 0 & 1 \\ 0 & 2 & 0 \end{bmatrix} = -0.5$$

$$\frac{\partial \lambda_{2r}}{\partial \eta} = [0 \quad 1] \begin{bmatrix} -0.5 & 0 & 0 \\ 0 & 2 & 1 \end{bmatrix} = 2$$

Thus, the eigenvalue sensitivities of the reduced-order model are identical with the corresponding sensitivities of the original model. On the other hand, with

$$R^{-1} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}$$

$$F = R^{-1} F_I R = \begin{bmatrix} 0 & -2 \\ 1 & -3 \end{bmatrix}, \quad v_{1r} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \quad v_{2r} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$w_{1r}^T = [1 \quad -1], \quad w_{2r}^T = [-1 \quad 2]$$

Hence,

$$\frac{\partial \lambda_{1r}}{\partial \eta} = [1 \quad -1] \begin{bmatrix} -0.5 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = -3$$

$$\frac{\partial \eta_{2r}}{\partial \eta} = [-1 \quad 2] \begin{bmatrix} -0.5 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 3.5$$

These sensitivities are not only different from the corresponding values for the original system but are also larger in magnitude.

Let us now consider the second-order eigenvalue sensitivities. For the original system,

$$\frac{\partial^2 \lambda_1}{\partial \xi^2} = w_1^T \left(2 \frac{\partial A}{\partial \xi} V p_1 + \frac{\partial^2 A}{\partial \xi^2} v_1 \right)$$

$$= [1 \quad -1 \quad 1] \left(2 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 3 & -3 & 1 \\ 5/2 & -4 & 3/2 \\ 1/2 & -1 & 1/2 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \\ -3/4 \end{bmatrix} + 0 \right) = -5/4$$

$$\frac{\partial^2 \lambda_2}{\partial \xi^2} = w_2^T \left(2 \frac{\partial A}{\partial \xi} V p_2 + \frac{\partial^2 A}{\partial \xi^2} v_2 \right)$$

$$= [1 \quad -2 \quad 4] \quad 2 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 3 & -3 & 1 \\ 5/2 & -4 & 3/2 \\ 1/2 & -1 & 1/2 \end{bmatrix} \begin{bmatrix} -1 \\ 0 \\ 3 \end{bmatrix} + 0 = -4$$

Hence,

$$\frac{\partial^2 F}{\partial \xi^2} = \begin{bmatrix} -5/4 & 0 \\ 0 & -4 \end{bmatrix}$$

For the case when R = I, $p_{12r} = (-2+1)\;[1\quad 0]\begin{bmatrix}-1/2 & 0\\ 0 & 2\end{bmatrix}\begin{bmatrix}0\\ 1\end{bmatrix} = 0$

$$p_{21r} = (-1+2)\;[0\quad 1]\begin{bmatrix}-1/2 & 0\\ 0 & 2\end{bmatrix}\begin{bmatrix}1\\ 0\end{bmatrix} = 0$$

Hence,

$$\frac{\partial^2\lambda_{1r}}{\partial\xi^2} = [1\quad 0]\;2\begin{bmatrix}-0.5 & 0\\ 0 & 2\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}0\\ 0\end{bmatrix} + \begin{bmatrix}-5/4 & 0\\ 0 & -4\end{bmatrix}\begin{bmatrix}1\\ 0\end{bmatrix} = -5/4$$

and

$$\frac{\partial^2\lambda_{2r}}{\partial\xi^2} = [0\quad 1]\;2\begin{bmatrix}-0.5 & 0\\ 0 & 2\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}0\\ 0\end{bmatrix} + \begin{bmatrix}-5/4 & 0\\ 0 & -4\end{bmatrix}\begin{bmatrix}0\\ 1\end{bmatrix} = -4$$

For the case when $R^{-1} = \begin{bmatrix}2 & 1\\ 1 & 1\end{bmatrix}$, $R = \begin{bmatrix}1 & -1\\ -1 & 2\end{bmatrix}$

$$p_{12r} = (-2+1)\;[1\quad -1]\begin{bmatrix}-0.5 & 0\\ 0 & 2\end{bmatrix}\begin{bmatrix}1\\ 1\end{bmatrix} = 2.5$$

$$p_{21r} = (-1+2)\;[-1\quad 2]\begin{bmatrix}-0.5 & 0\\ 0 & 2\end{bmatrix}\begin{bmatrix}2\\ 1\end{bmatrix} = 5$$

$$\frac{\partial^2\lambda_{1r}}{\partial\xi^2} = [1\quad -1]\;2\begin{bmatrix}-0.5 & 0\\ 0 & 2\end{bmatrix}\begin{bmatrix}2 & 1\\ 1 & 1\end{bmatrix}\begin{bmatrix}0\\ 2.5\end{bmatrix} + \begin{bmatrix}-3 & 0\\ 0 & -4\end{bmatrix}\begin{bmatrix}2\\ 1\end{bmatrix} = 20.5$$

$$\frac{\partial^2\lambda_{2r}}{\partial\xi^2} = [-1\quad 2]\;2\begin{bmatrix}-0.5 & 0\\ 0 & 2\end{bmatrix}\begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix}\begin{bmatrix}5\\ 0\end{bmatrix} + \begin{bmatrix}-3 & 0\\ 0 & -4\end{bmatrix}\begin{bmatrix}1\\ 1\end{bmatrix} = 45$$

6. CONCLUSIONS

Expressions for first- and second-order eigenvalue sensitivities of aggregated models with respect to the parameters of the original system have been calculated. These expressions turn out to be quite simple in form and show the relationships between the eigenvalue sensitivities of the model and the original sensitivities. In particular, if the aggregated model is in the diagonal form, then its eigenvalue sensitivities are identical with the corresponding eigenvalue sensitivities in the original high-order system.

It may be possible to select the aggregation matrix in such a manner that the eigenvalue sensitivities in the reduced model are minimized. This is a subject for further research.

7. ACKNOWLEDGEMENTS

The authors are grateful to J. Hickin for very interesting discussions.

The financial support of the National Research Council of Canada is gratefully acknowledged.

8. REFERENCES

1. M. Aoki, "Control of large scale systems by aggregation", IEEE Trans. on Automatic Control, vol. AC-13, 1968, pp. 246-253.

2. J. Hickin and N.K. Sinha, "Projective methods for reducing linear dynamic systems and applications to estimation and control", Proc. 13th Annual Allerton Conference on Circuit and System Theory, 1975, pp. 39-47.

3. E.J. Davison, "A method for simplifying linear dynamic systems", IEEE Trans. on Automatic Control, vol. AC-11, 1966, pp. 246-253.

4. D. Mitra, "Analytical results in the use of reduced models in the control of linear dynamic systems", Proceedings I.E.E., vol. 116, 1969, pp. 1439-1444.

5. G. Michailesco, J.M. Siret and P. Bertrand, "Aggregated models for high-order systems", Electronics Letters, vol. 11, 1975, pp. 398-39.

6. J. Hickin and N.K. Sinha, "Optimally aggregated models of high-order systems", ibid., vol. 11, 1975, pp. 632-633.

7. J. Hickin and N.K. Sinha, "Singular perturbations and aggregation", Proc. 8th Annual Pittsburgh Conference on Modeling and Simulation, April 1977.

8. P.V. Koktovic and P. Sannuti, "Singular perturbation method for reducing the model order in optimal control design", IEEE Trans. on Automatic Control, vol. AC-13, 1968, pp. 377-384.

9. H.M. Zein El-Din, R.T.H. Alden and P.C. Chakravarti, "Second-order eigenvalue sensitivities applied to multivariable control systems", Proc. IEEE, vol. 65, 1977, pp. 277-278.

THE NESTED BORDERED-DIAGONAL METHOD FOR SYMMETRIC EIGENPROBLEMS

PETER P. GROUMPOS, PETER D. SCOTT

SUNY at Buffalo, Dept. of Electrical Engineering
Amherst, New York 14261

ABSTRACT

Numerical evaluation of all the eigenvalues and eigenvectors of a real symmetric matrix is considered. The nested bordered-diagonal method is developed and compared with existing algorithms. The eigenvalues and eigenvectors are computed by preprocessing the matrix to a "filled-arrow" form, followed by nested diagonalizations of bordered-diagonal ("arrow") partitions. The relationship between the arrow form and hierarchical structures of general and large-scale systems is discussed.

I. INTRODUCTION

Hermitian matrices are frequently encountered in practice, usually in the form of real symmetric matrices. The problem of numerical evaluation of all the eigenvalues and eigenvectors of such matrices is one of the most common in applied mathematics. Their algebraic properties have been frequently explored.

The Nested Bordered-Diagonal (NBD) method presented here employs a finite series of coordinate transformations to produce a "filled arrow" form, then through nested diagonalization of "arrow" or bordered-diagonal partitions, the eigenvalues and eigenvectors of a real symmetric matrix are determined. This method combines ideas drawn from coordinate transformation and tridiagonal techniques [4], [15] with those from (currently unfashionable) escalator methods [5] for sequential bordered diagonalization of leading principal minors.

The organization of the paper is as follows. In the next section, existing algorithms for the eigenproblem are discussed, with an emphasis in the symmetric case.

In section 3 the basic steps of the NBD method are briefly discussed. The use of preprocessing techniques is outlined and the notion of the "arrow" form of a matrix is introduced. Through the NBD method all eigenvalues and eigenvectors of a real symmetric matrix are determined.

In section 4, we introduce the arrow form of a real symmetric matrix and describe how to compute its eigenvalues and corresponding eigenvectors.

In section 5, computational considerations are discussed and numerical examples are presented. Comparisons with the QR algorithms are presented.

Finally, in section 6 conclusions are drawn from the NBD approaches in system analysis.

II. THE EIGENPROBLEM

The problem of eigenvalue computation is an old one. In the last several years there has been considerable effort in developing and analyzing algorithms for computing the eigenvalues of a matrix A [2]-[8], [10], [15].

Methods for symmetric matrices include the Givens method (GM) [8] and the Householder method (HM) [10], [11], [15]. Both methods employ orthogonal similarity transformations (OST) to modify the given matrix $\underline{A}$ to a tridiagonal form, whose eigenvalues and eigenvectors are more easily computed [4], [15].

Another algorithm is the well known QR-algorithm [6], [7]. This algorithm is based on unitary similarity transformations. It is based on reducing the matrix to an almost triangular matrix, for the symmetric case to a tridiagonal one, and then through an infinite sequence of factorization diagonal form is obtained. The diagonal elements are the eigenvalues of the original matrix.

For the nonsymmetric case the given matrix $\underline{A}$ is transformed to the Hessenberg form through a serious of elementary transformations. Then various methods are employed to find the eigenvalues and eigenvectors of the Hessenberg form. We mention here the Jacobi method and Muller's method [4].

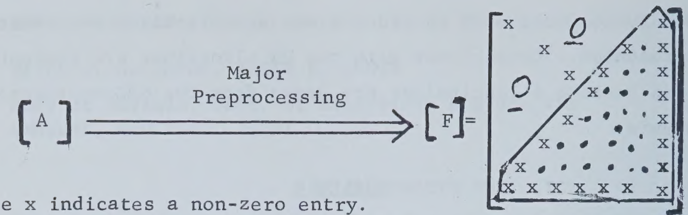
None of the above algorithms are finitely convergent and a currently interesting question is how well do they do as a function of the dimensionality of the system, when size and sparseness becomes inherent features of the overall system.

III. THE NESTED BORDERED-DIAGONAL (NBD) APPROACH

Suppose we are given a real symmetric matrix $\underline{A}$ of dimensionality n x n and wish all the eigenvalues and the corresponding eigenvectors.

Preprocessing techniques have been used in various areas, particularly on networks, producing very favorable results [3], [14].

A major preprocessing scheme appropriate to bordered diagonalization consists of reduction of $\underline{A}$ to the "filled-arrow" matrix form $\underline{F}$:



(3-1)

where x indicates a non-zero entry.

To obtain the above form the following algorithm may be used [9].

1) Create zeros using elementary orthogonal similarity transformations (EOST) on a row, say the i^{th} row, for the entries a_{ij}, $j = i+1, i+2 \ldots n-i$. The pivoting vectors are the j^{th} and $n-i+1$ row and column respectively.

2) Repeat (1) for
   (i) even case $i = 1,2,\ldots n/2$
   (ii) odd case $i = 1,2,\ldots n-1/2$

Note 1 Due to the EOST zeros will be also created in the i^{th} column.

Note 2 If the operations are done in row order no zeros are destroyed by later operations. Thus, one rotation per nonzero symmetric pair is required.

When the above algorithm is used as a major preprocessing scheme the following forms $\underline{F_1}$, $\underline{F_2}$ can be obtained.

a) Even case (e.g. n = 6) b) odd case (e.g. n = 7)

$$F_1 = \begin{bmatrix} x & 0 & 0 & 0 & 0 & x \\ 0 & x & 0 & 0 & x & x \\ 0 & 0 & x & 0 & x & x \\ 0 & 0 & 0 & x & x & x \\ 0 & x & x & x & x & x \\ x & x & x & x & x & x \end{bmatrix} \qquad F_2 = \begin{bmatrix} x & 0 & 0 & 0 & 0 & 0 & x \\ 0 & x & 0 & 0 & 0 & x & x \\ 0 & 0 & x & 0 & x & x & x \\ 0 & 0 & 0 & x & x & x & x \\ 0 & 0 & x & x & x & x & x \\ 0 & x & x & x & x & x & x \\ x & x & x & x & x & x & x \end{bmatrix}$$

Remark

Note that in the even case two zeros are created on the secondary diagonal, where that is not the case in the odd case.

A related form $\underline{\hat{A}}$, derived from F_1 or F_2 by similarity transformations, eliminates all non-zero entries except on the main diagonal and last row/column:

$$\hat{A} = \begin{bmatrix} x & & & \underline{0} & & x \\ & x & & & & x \\ & & x & & & x \\ \underline{0} & & & \cdot & & x \\ & & & & \cdot & x \\ x & x & x & x & x & x \end{bmatrix} \tag{3-2}$$

$\hat{\underline{A}}$ will be referred to as the arrow form of $\underline{A}$, and F_1 and F_2 the related filled-arrow forms.

The basic steps in the NBD method can be summarized for the odd case as follows: (The details for n even are similar but only n odd is assumed in this paragraph)

(i) Preprocessing: the given n x n real symmetric matrix $\underline{A}$ is brought to the filled arrow form through a series of $\left(\frac{N-1}{2}\right)^2$ rotations (elementary orthogonal similarity transformations)

(ii) The central 3 x 3 partition of the filled arrow form is a symmetric arrow form submatrix $\underline{A}_3$. The model matrix x_3 and diagonalized 3 x 3 form D_3 $(x_3^T A_3 x_3 = D_3)$ are found as in section IV.

(iii) The central coordinate transformation $A' = P_3^T F P_3$ with

$$P_3 = \left[\begin{array}{c|c|c} I_{\left[\frac{N}{2}\right]-1} & 0 & 0 \\ \hline 0 & x_3 & 0 \\ \hline 0 & 0 & I_{\left[\frac{N}{2}\right]-1} \\ \hline & & \end{array}\right] \qquad (3\text{-}3)$$

yields A′ with central 5 x 5 partition in symmetric arrow form.

i) repeat ii) and iii) increasing dimensions of central arrow form by two and determine the appropriate central coordinate transformation P until the entire n x n matrix is diagonalized, and the eigenvalues of $\underline{A}$ are the diagonal elements.

IV. THE ARROW FORM OF A REAL SYMMETRIC MATRIX AND SOME OF ITS PROPERTIES

In the previous section we defined the arrow form $\hat{\underline{A}}$ of a given matrix $\underline{A}$. The arrow form of a real symmetric matrix is also of the form (3-2) with the additional fact that it is symmetric. Let us write $\hat{A}$ for the symmetric case as follows:

$$\hat{A}_n = \begin{bmatrix} \gamma_1 & 0 & 0 & \cdot & \cdot & \cdot & c_1 \\ 0 & \gamma_2 & 0 & \cdot & \cdot & \cdot & c_2 \\ 0 & 0 & \gamma_3 & \cdot & \cdot & \cdot & c_3 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ c_1 & c_2 & c_3 & \cdot & \cdot & \cdot & g \end{bmatrix} \qquad (4\text{-}1)$$

We state the following properties associated with (4-1). Details are given in [9] and [13].

Theorem 4.1 The characteristic equation of (4-1), is

$$f_n(\lambda) = - \sum_{j=1}^{n-1} c_j^2 \prod_{\substack{i=1 \\ i\neq j}}^{n-1} (\lambda-\gamma_i) + (\lambda-g) \prod_{i=1}^{n-1} (\lambda-\gamma_i) \tag{4-2}$$

Corollary 4.1 The characteristic equation $f_n(\lambda)$ of (4-1) can be evaluated by the recursions.

$$p_{k-1}(\lambda) = (\lambda-\gamma_{k-1})p_k(\lambda) - c_{k-1}^2 q_k(\lambda)$$

$$q_{k-1}(\lambda) = q_k(\lambda)(\lambda-\gamma_k) \tag{4-3}$$

for k = n, n-1, . . .1 with boundary final values $p_n(\lambda) = \lambda-g$, $q_n(\lambda) = 1$; and $f_n(\lambda) = p_0(\lambda)$.

Note that for k = n, $\gamma_n = g$.

Observation: The coefficients of the characteristic polynomial need not be explicitly determined (also a significant feature in tridiagonal algorithms), if the recursions are evaluated for even a new λ.

Lemma 4.1 For $\underline{A}$ and $\underline{B}$ real symmetric matrices with $\underline{B}$ non-negative definite then

$$\lambda_k(A + B) \geq \lambda_k(A) \qquad k = 1,2, . . . n \tag{4-4}$$

where λ_k indicates the k^{th} ordered eigenvalue of A; If $\underline{B}$ is positive definite then (4-4) is a strict inequality.

Lemma 4.2 Let $\underline{B}$ and $\underline{C}$ be real symmetric matrices and hence so is $\underline{A}$, A = B + C then,

(i) the eigenvalues of $\underline{A}$ are both equal to the corresponding eigenvalues of $\underline{B}$ perturbed by at most the largest and smallest eigenvalues of $\underline{C}$.

(ii) the eigenvalues of the leading principal minor $\underline{A}_{n-1}$ of the matrix $\underline{A}$ separate the eigenvalues $\lambda_1, \lambda_2, . . . \lambda_n$ of A .

The proofs of both lemmas are in Wilkinson [15].

Theorem 4.2 Let the arrow form (4-1) be arranged such that $\gamma_1 \geq \gamma_2 \geq \gamma_3 \geq . . . \gamma_{n-1} \geq g$. Then the spectrum of A_n satisfies

$$\sqrt{c^T c} + \gamma_1 \geq \lambda_1 \geq \gamma_1 \geq \lambda_2 \geq \gamma_2 \; . \; . \; . \geq \lambda_{n-1} \geq g \geq \lambda_n \geq g - \sqrt{c^T c} \qquad (4\text{-}5)$$

where $\underline{c}$ is the last column of $\underline{A}_n$ excluding g.

Proof in Appendix.

Furthermore, effective bounds on the location of each eigenvalue of A_n are determined through conjunction of the following three facts:

a) $|\lambda_i - \gamma_i| \leq |c_i|$ for $i = 1,2 \; . \; . \; . n-1$, i.e. the perturbation in the i^{th} eigenvalue created by bordering the diagonal $(n-1)x(n-1)$ leading minor of A_n by its last row and column is limited to the single coupling coefficient c_i $(i = 1,2 \; . \; . \; . n-1)$

b) all perturbation sum to zero, i.e. the n^{th} eigenvalue may be determined by the first n-1, $\lambda_n = g + \sum_{i=1}^{n-1} (\gamma_i - \lambda_i)$ and

c) the eigenvalues of A_n are separated by the numbers $\gamma_1, \gamma_2 \; . \; . \; . \gamma_{n-1}$

Now the n-1 eigenvalues of A_n may be determined by n-1 one dimensional searches within the above mentioned bounds a), using (4-3) to evaluate the the characteristic polynomial. A very effective one dimensional search procedure for this problem consists fitting a quadratic to the characteristic equation by evaluation at the two bounds and at an intermediate point. Since repeated roots always corresponds to zero components of the c-vector in the arrow form [9], these can be factored and the value of the reduced characteristic equation will be of opposite sign at the two bounds. Only a single zero, both in the characteristic equation and in the quadratic model, will exist in the bounded subinterval. This guarantees quadratic convergence if the direct interval elimination scheme which results from evaluating the characterisitc equation on the computed zero of the quadratic model.

In a very simple and nice way, by inspection, the corresponding eigenvectors of A_n can be found. For the case of distinct eigenvalues for a particular eigenvalue λ_i of A_n, we want to find x_i such that

$$(\lambda_i I - A_n)x_i = 0 \qquad (4\text{-}5)$$

or

$$\begin{bmatrix} k_1 & 0 & 0 & . \; . \; . & -c_1 \\ 0 & k_2 & 0 & . \; . \; . & -c_2 \\ 0 & 0 & k_3 & . \; . \; . & -c_3 \\ . & . & . & . \; . \; . & . \\ . & . & . & . \; . \; . & . \\ -c_1 & -c_2 & -c_3 & . \; . \; . & k_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ . \\ x_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ . \\ . \\ 0 \end{bmatrix} \qquad (4\text{-}6)$$

where $k_1, k_2 \ldots k_{n-1}$ are the ordered values $\lambda_i - \gamma_i$ and $k_n = \lambda_i - g$. x_n must be nonzero, so let $x_n = 1$, and $k_1 x_1 = c_1 = x_i = c_1/k_1$. Proceeding similarly for the rest of x_i we have the following corresponding eigenvectors $x_i = [c_1/k_1 \; c_2/k_2 \ldots c_{n-1}/k_{n-1} \; 1]^T$ (k_i cannot be zero [9]).

If we have repeated eigenvalues, from lemma 4.2 and theorem 4.2 we see that if γ_i is repeated M+1 times, then its order as an eigenvalue is exactly M (assuming all $c_i \neq 0$). Since any real symmetric matrix can be diagonalized there are M linearly independent eigenvectors associated with these M repeated eigenvalues.

According to the values of c_i and k_i of equation (4-6), there are various cases which determine the corresponding eigenvectors for the distinct as well as the repeated cases and they have been covered in [9] and [13].

In this section we have shown how the eigenvalues and eigenvectors of the arrow form of a real symmetric matrix can be computed.

V. NUMERICAL CONSIDERATIONS

The NBD method has been implemented on the university's CDC Cyber 173 computer and compared to a QR code.

The computer program is composed of the main program, several subroutines and a function subprogram. The various subroutines are implementing the major preprocessing techniques to bring the real symmetric matrix to a filled arrow form, finding the eigenvalues of an arrow form and then the corresponding eigenvectors, rearrange the diagonal elements of a principal minor in a descending order and perform one-dimensional quadratically convergent search. The function subprogram is simply the evaluation $f_n(\lambda) = p_o(\lambda)$ from eqn. (4-3).

Results to date show comparable execution times and accuracy for most problems up to 20 x 20. In several trials the QR algorithm failed because of poor conditioning or closely spaced eigenvalues, while the NBD algorithm showed no such sensitivities

Example 1

Suppose we have the 6 x 6 matrix A

$$A = \begin{bmatrix} 5 & 3 & 2 & 1 & 3 & 4 \\ 3 & 5 & 2 & 8 & 2 & 1 \\ 2 & 2 & 5 & 6 & 4 & 2 \\ 1 & 8 & 6 & 5 & 3 & 1 \\ 3 & 2 & 4 & 3 & 9 & 2 \\ 4 & 1 & 2 & 1 & 2 & 14 \end{bmatrix}$$

Results are summarized on the following table.

TABLE 1.1

	QR	NBD
Eigenvalues	22.067471	22.0674711
	13.032058	13.032059
	6.935626	6.935634
	3.676739	3.675739
	1.718940	1.718940
	-4.429835	-4.429835
Execution Time	0.1730 sec.	0.2106

Example 2

In a 10 x 10 real symmetric matrix, same accuracy was sustained for the eigenvalues. The execution time was

QR algorithm = 0.776 sec.
NBD algorithm = 0.539 sec.

For the 6 x 6 and 10 x 10 cases many different matrices were tested with similar results. Although a 20 x 20 case was tried, due to convergence failure of the QR code we cannot present any reliable judgment on computation times between the two methods. The NBD method gave the correct eigenvalues with computation time of 2.8 seconds.

Note 1 No attempt has been made to optimize either code.

Note 2 The QR algorithm does not give ordered eigenvalues where the NBD algorithm does.

VI. CONCLUSIONS

This study was motivated by our interest in developing techniques which at each iteration simultaneously assesses the strength of interaction between variables or subsystems while progressing toward a solution. By bringing into the algorithm only those variables or subsystems which interact most strongly with one another, a natural decomposition and approximation procedure has been developed which at each step reveals the most accurate "form" of reduced structure for the system. This has been done in detail for the eigenvalue problem for the symmetric matrix, by analyzing a matrix, whose the leading principal minor is diagonal, referred to as the arrow form.

As a result of this study, the NBD method was developed as an alternative to compute all the eigenvalues and eigenvectors of a real symmetric matrix. The method appears competitve with existing techniques for the symmetric eigenproblem and combines ideas from several camps in a novel way.

The arrow form matrix appears in many hierarchical and large-scale contexts and deserves further related analysis. We find this is a natural form for development of hierarchical control strategies. The more difficult non-symmetric case needs also to be considered.

ACKNOWLEDGEMENT

We would like to thank Professor D. Kazakos for his encouragement and help during the time this research was carried out.

REFERENCES

1. R.E. Bellman, Introduction to Matrix Analysis, McGraw Hill, 1960.
2. D.A. Callahan and W.J. McCalla "Eigenvalue Methods for Sparse Matrices," Proceedings of a Symposium on Sparse Matrices and their Applications, IBM, J.J. Watson Research Center, September 1971.
3. I. Chandrashekar, "Computer Analysis of Large-Scale Systems," Ph.D. Thesis, Hawaii University.
4. L. Fox, An Introduction to Numerical Linear Algebra, Claredon Press, Oxford, 1964.
5. L. Fox, "Escalator Methods for Latent Roots," Quart J., Mech. 5, pp. 178-90, 1952.
6. J.G.F. Francis, "The QR Transformation -A Unitary Analogue to the LR Transformation-Part I," Computer J., 4, pp. 265-271, 1961.
7. J.G. F. Francis, "The QR Transformation-Part II," Computer J., 4, pp. 332-345, 1962.
8. W. Givens, "A Method of Computing Eigenvalues and Eigenvectors Suggested by Classical Results on Symmetric Matrices," App. Math. Sci. Nat. Bur. Stand. 29, pp. 117-122, 1953,
9. P.P. Groumpos, "Bordered Diagonal Matrices in Linear Systems Analysis," Master's Thesis, SUNYAB, 1976.
10. S.J. Hammarling, Latent Roots and Latent Vectors, The University of Toronto Press, 1970.
11. A.S. Householder, The Theory of Matrices in Numerical Analysis, Blaisdell Publishing Company, 1964.
12. M.D. Mesarovic, D. Mouk and Y.T. Akahara, Theory of Hierarchical Multilevel Systems, Academic Press, 1970.
13. P.D. Scott and P.P. Groumpos, "A New Algoirthm for Eigenvalues Calculation," EE Dept., SUNYAB, Technical Report, (in preparation).
14. W.F. Tinney and J.W. Walker, "Direct Solutions of Sparse Network Equations by Optimally Ordered Triangular Factorization," IEEE Proceed, vol. 55, pp. 1801-1809, November 1967.
15. J.H. Wilkinson, The Algebraic Eigenvalue Problem, Claredon Press, Oxford, 1965.

APPENDIX

Proof of Theorem 4.2

Let $A_n = B + C$

where

$$B = \begin{bmatrix} \gamma_1 & 0 & 0 & . & . & . & 0 \\ 0 & \gamma_2 & 0 & . & . & . & 0 \\ 0 & 0 & \gamma_3 & . & . & . & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & . & . & . & g \end{bmatrix}$$

$\gamma_1 \geq \gamma_2 \geq \gamma_3 \ . \ . \ . \geq g$

and

$$C = \begin{bmatrix} 0 & 0 & 0 & . & . & . & c_1 \\ 0 & 0 & 0 & . & . & . & c_2 \\ 0 & 0 & 0 & . & . & . & c_3 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ c_1 & c_2 & c_3 & . & . & . & 0 \end{bmatrix}$$

The eigenvalues of B are $\gamma_1, \gamma_2, . . . g$. The eigenvalues of C can be determined using theorem 4.1.

$$f(\lambda) = \lambda^n - \sum_{i=1}^{n-1} c_i^2 \lambda^{n-2} \qquad (1)$$

or

$$f(\lambda) = \lambda^{n-2} [\lambda^2 - \sum_{i=1}^{n-1} c_i^2] \qquad (2)$$

From (2) we see that n-2 eigenvalues of C are zero. Furthermore, the other two ones can be easily found

$$\lambda^2 - \sum_{i=1}^{n-1} c_i^2 = 0$$

Let

$$c^T c = \sum_{i=1}^{n-1} c_i^2$$

Hence the two roots are

$$\lambda max = \sqrt{c^T c}$$

and

$$\lambda min = -\sqrt{c^T c}$$

So theorem 4.2 is a direct result of lemma 4.2.

NORMALIZATION OF SYSTEM VARIABLES AND MULTIVARIABLE SYSTEM ANALYSIS[*]

RATKO PLAČKOVIĆ[**]
Decision and Control Laboratory
Coordinated Science Laboratory
University of Illinois
Urbana, Illinois 61801

ABSTRACT

Effects of the normalization of plant model on the system analysis for the purpose of control system design are investigated. It is shown that normalization is essential for evaluation of structural properties of industrial process control systems. A convenient normalization proposed in this paper is based on using given or defined maximal available perturbations of the system variables in the range of the plant operation. The procedure is illustrated on a boiler-turbine unit.

INTRODUCTION

Design of complex industrial process control systems requires precise definition of control requirements and systematic analysis of plant properties in order to specify control structure and strategies. These requirements stem from interactions of the plant with the external system and from requirements for the safe and reliable operation of the plant and are usually formulated as (a) minimizing of input energy and material expenditure while delivering the process product, i.e., plant output of specified quality, and (b) maintaining the process variables inside prescribed limits. In process control systems individual controls are generally not functions of all available measurements. In such systems each output is primarily influenced by certain of the system modes. Qualitatively speaking, one is interested in defining a control structure in which the controls are functions of the outputs which are most influenced by changes in those modes most affected by the controls. Clearly, a selection of the control structure requires analysis of the mutual influences of the process variables. Thus, it depends also on the criteria used to judge the levels of these influences. Moreover, the

---

[*] This work was supported by the Energy Research and Development Administration, Electric Energy Systems Division, under Contract U. S. ERDA EX-76-C-01-2088.

[**] On leave 1977 from Electrotechnical Institute Rade Končar, Zagreb, Yugoslavia.

result of such an analysis will generally not yield a unique control structure but rather a class of structures. A specific choice of the control structure from this class will depend on further system analysis and on the constraints dictated by available control system realizations in the particular case.

The result of industrial process modeling is usually a multivariable nonlinear model. This model is then linearized around the operating point(s) of interest and the resulting linear system is used to analyze the plant properties.

In this paper we wish to point out and illustrate on an example (a boiler turbine unit) that the linear model so obtained is not necessarily a satisfactory representation of the system for specifying the appropriate control structure. The system variables are given in different physical units and their range of changes depends on the plant operating point. They should be normalized with respect to values having properties which are common to all the variables' perturbations around the equilibrium point. Properties to be used should be chosen so that the effects of changes in the resulting normalized variables on the state of the system are easy to compare or the use of the results of analysis could lead to wrong conclusions. The importance of proper normalization is also recognized in [1] where observability of a system was considered and r.m.s. values of the state variables were proposed for their normalization. The process variables are not always characterized by their r.m.s. values and this choice would not be suitable for the class of systems considered here.

Our intention is to show that the aforementioned requirements for normalization are satisfied if the available maximal perturbations of the system variables in the plant operating range under consideration are used for normalization. The influence of the normalization on the system analysis is investigated and illustrated on the example.

MODELING

Process modeling based on laws of conservation of mass, energy and momentum usually results in a system of nonlinear differential and algebraic equations in the form

$$\begin{aligned} \dot{x}_b &= f(x_b, u_b, z_b) \\ 0 &= g(x_b, u_b, z_b) \end{aligned} \qquad (1)$$

where $f(x_b,u_b,z_b)$ and $g(x_b,u_b,z_b)$ are time-invariant vector valued nonlinear functions with $x_b \in R^n$ the state variable, $u_b \in R^m$ control and $z_b \in R^\ell$ measured output. If f and g are continuous and differentiable with respect to x_b, u_b, and z_b for the range of the plant operation under consideration, the model can be linearized around a chosen operating point by using a Taylor series expansion of f and g at the nominal operating point and neglecting second and higher order terms in the expansions. Let the operating point be (x_b^o,u_b^o,z_b^o) and corresponding perturbations be

$$x_p = x_b - x_b^o, \quad u_p = u_b - u_b^o, \quad z_p = z_b - z_b^o. \tag{2}$$

The resulting linear model is then

$$\dot{x}_p = A_p x_p + B_p u_p \tag{3a}$$

$$z_p = H_p x_p + F_p u_p \tag{3b}$$

where matrices $A_p \in R^{n\times n}$, $B_p \in R^{n\times m}$, $H_p \in R^{\ell\times n}$, and $F_p \in R^{\ell\times m}$ are functions of x_b^o, u_b^o, and z_b^o. If a requirement on the control system is to regulate to the desired values the vector $y_b \in R^r$ which is linearly related to z_b, then the perturbations of y_b are given by

$$y_p = C_p x_p + D_p u_p \tag{3c}$$

where matrices $C_p \in R^{r\times n}$ and $D_p \in R^{r\times m}$ are defined by the linear relation between z_p and y_p and by (3b). Therefore the system can be represented by (3a) and (3c) and y_p used as the output.

In this paper a reheat drum type coal fired boiler-turbine unit with rated turbine power output of 230 MW [2] will be used to illustrate the points under consideration. The requirements on the control system associated with this plant are to provide maximal rate of change of the output power generation from the initial to a target equilibrium point without excessive energy expenditure while maintaining the process variables (Appendix A1) inside prescribed limits. The considered range of operation corresponds to output generation between 200 MW and 220 MW. Safe and reliable operation requires maintaining the plant outputs inside the limits given in Appendix A1. The control constraints given by physical realization of the plant actuators are listed in A1.

The boiler turbine-unit dynamic properties are described by the equations derived from the corresponding laws governing the transfer of energy and mass and the equations of state describing material properties. The mathematical model is nonlinear and of 14th order [3]. The measured output variables in the model are output power generation (z_1), reheater outlet temperature (z_2), average reheater temperature (z_3), secondary superheater outlet temperature (z_4), average secondary superheater temperature (z_5), throttle pressure (z_6), drum pressure (z_7), throttle flow (z_8), feedwater flow (z_9) and drum water level (z_{10}). The output signal used to control the drum water level in the existing control system is

$$y_8 = z_8 - z_9 - cz_{10} \qquad (4)$$

where c is a constant found by experiments to be 0.28 [4], and this output signal is used in this work as well. The other output signals to be controlled are

$$y_p^i = z_p^i, \qquad (i = 1,2,\ldots,7). \qquad (5)$$

The controls and state variables in the model are defined as given in Tables 2 and 3 in Appendix A1, respectively.

The model is linearized around the operating point corresponding to an output generation of 210 MW. The equilibrium values of the model variables are given in A1.

NORMALIZATION

System (3) is not the most convenient for analysis of interactions between the system variables. Generally, the process variables are given in different physical units and cannot change from zero to the nominal values in the range of operation under consideration. Their available changes depend on the control system requirements, actuator properties and physical properties of the plant. For example, the requirements on the boiler-turbine unit for the plant operation range under consideration [4] are (among others) to maintain the throttle pressure at 1825 psia with tolerance $\pm$ 50 psia and throttle temperature at 1000°F with tolerance $\pm$ 15°F. The superheater and reheater attemperator spray flows can change from zero to 0.0167 klb/s and not more. The fuel input is changing correspondingly to the change of energy input and coal mill feeder stroke will not change by more than 10% of the maximal value. Therefore, available perturbations of these process variables around the nominal point are 100 psia, 30°F, 0.0167 klb/sec and 10% respectively. Hence, the control system when applied to the plant will deal with the disturbances in the range of system operation given by the maximal available perturbations of the process variables corresponding to the operating point under consideration. The controller will receive information about the system operation (represented by x_p) through y_p and affect actuators by u_p if corrective action will be required. We wish to have each component of u_p as a function of not all but some of the outputs y_p, and to select a control system structure in which we have to associate individual outputs to individual controls. When coupling between system variables is strong, we have to analyze the influences of changes of each of the system variables on all the other variables. To compare the levels of these influences it is necessary to have the variables represented in such a way that their changes are easy to compare. A convenient representation of the changes in y_p, x_p, and u_p is given by the ratio of these variables and their respective maximal available range of perturbations.

In this paper the maximal available perturbations of the system variables for the range of operation corresponding to the nominal operating point under consideration are used for normalization of the model variables. They are, for convenience named the nominal perturbations. The notion of the nominal perturbations enables a restatement of the requirements on the control system.

The constraints on the output variables can be expressed as

$$-y_n^i/2 \le y_p^i \le y_n^i/2 \qquad (i = 1,2,\ldots,r) \tag{6}$$

where y_n^i are the nominal output perturbations. The nominal output perturbations define the r-dimensional parallelepiped (generally, polyhedron) in output space. For the case $r = 2$ this polyhedron would be a rectangle (Fig. 1). The control variable constraints are also known and are given by

$$-u_n^i/2 \le u_p^i \le u_n^i/2 \tag{7}$$

for $i = 1,2,\ldots,m$, where u_n^i are nominal control perturbations. They define the m-dimensional polyhedron M_u in the control space. State variable constraints are not explicitly defined but can be evaluated by simulation and/or engineering judgement. The resulting constraints define the nominal state variable perturbations x_n^i, $i = 1,2,\ldots,m$

$$-x_n^i/2 \le x_p^i \le x_n^i/2 \qquad (i = 1,2,\ldots,n). \tag{8}$$

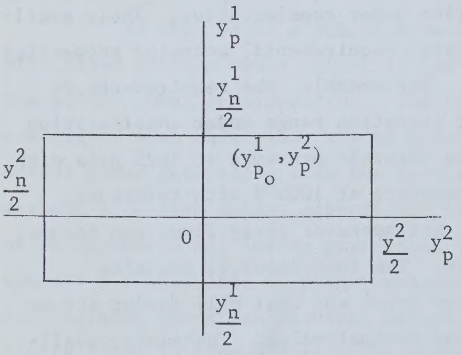


Fig. 1

The nominal state variable perturbations define the n-dimensional polyhedron M_x in the state space. Control requirements can now be reformulated as: the controller should generate controls in M_u to steer the system from the initial point to a target point along the trajectory in M_x (without excessive expenditure of input energy and material) which provide the required trajectory (trajectory as close as possible to the required trajectory) in M_y.

When the system is normalized with respect to the nominal perturbations, the corresponding polyhedrons M_u, M_x, and M_y have edges of unit length. The maximal possible change of any system variable is 1, and thus it is easy to compare the effects of the system variable changes on all other system variables.

The normalization algorithm is straightforward. Nominal system variable perturbations are used to form diagonal normalization matrices. The normalization matrices are $X^n = \{X^n_{i,j}\} \in R^{n\times n}$, $Y^n = \{Y^n_{i,j}\} \in R^{r\times r}$, and $U^n = \{U^n_{i,j}\} \in R^{n\times m}$, such that $\mathrm{diag}(X^n) = \{x^i_n\}$, $i = 1,2,\ldots,n$, $\mathrm{diag}(Y^n) = \{y^i_n\}$, $i = 1,2,\ldots,r$ and $\mathrm{diag}(U^n) = \{u^i_n\}$, $i = 1,2,\ldots,m$ with offdiagonal elements all zero. Let

$$x_p = X^n x, \quad y_p = Y^n y, \quad u_p = U^n u. \tag{9}$$

The normalization of the system (3) gives

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{10}$$

where

$$A = (X^n)^{-1} A_p X^n, \quad B = (X^n)^{-1} B_p U^n, \quad C = (Y^n)^{-1} C_p X^n, \quad D = (Y^n)^{-1} D_p U^n. \tag{11}$$

The nominal perturbations of the system variables and the normalized matrices A, B, C, and D of the boiler-turbine unit model are given in Appendices A1 and A2, respectively.

ANALYSIS OF SYSTEM STRUCTURAL PROPERTIES AND NORMALIZATION

The approach used in investigating system structural properties used in this work is based on modal form of the system (10).

$$\dot{z} = \Lambda z + T^{-1}Bu \qquad y = CTz + Du \tag{15}$$

where $z = T^{-1}x$ is the modal state variable, Λ is a diagonal matrix with $\mathrm{diag}(\Lambda) = [\lambda_1\ \lambda_2 \ldots \lambda_n]$, λ_i are eigenvalues of A, $T \in R^{n\times n}$ is the modal transformation matrix (with column vectors normalized with respect to their Euclidean norms).

The modal observability matrix $Q_o = \{q^o_{i,j}\} \in R^{r\times n}$ and the modal controllability matrix $Q_c = \{q^c_{i,j}\} \in R^{n\times m}$ are defined as

$$Q_o = CT \quad \text{and} \quad Q_c = T^{-1}B. \tag{16}$$

The values of the elements of Q_o and Q_c can be used as measures of observability and controllability of system (10) variables [1,5]. For example, the value of the element $q^c_{i,j}$ can be used as a measure of the controllability of the ith modal state variable by the jth control input. The matrices Q_o and Q_c depend on the values used for normalization of the system variables. To show this, let $X^a \in R^{n\times n}$, $Y^a \in R^{r\times n}$, and $U^a \in R^{n\times m}$ be nonsingular diagonal matrices defined by the set of values $\{a_i\}$, $i = 1,2,\ldots,n+r+m$ different than the set of nominal perturbations. Then, if X^a, Y^a, and U^a are used for normalization the corresponding modal observability and controllability matrices would be

$$\begin{aligned} Q^a_c &= T^{-1}(X^a)^{-1}B_pU^a \\ \text{and} \qquad Q^a_o &= (Y^a)^{-1}C_pX^aT. \end{aligned} \tag{17}$$

Comparing (16) with (17), respectively, it can be easily seen that $Q_o \neq Q^a_o$ and $Q_c \neq Q^a_c$. Therefore, the proper choice of the normalization values is essential for the analysis of the system structural properties.

The eigenvalues corresponding to the linearized model of the boiler-turbine unit are given in Appendix A3. The first eigenvalue (which corresponds to the drum water volume) is zero and hence the linearization theorem gives no answer as to whether the plant is stable or not. The assumption based on the physical considerations is that the considered equilibrium is stable for any (physically reasonable) set of steady state values of the controls. The modes corresponding to the eigenvalues λ . $i = 2,3,\ldots,14$ are stable.

The boiler-turbine unit modal observability and controllability matrices Q_o and Q_c are given in Appendix A3. The matrices Q^a_o and Q^a_c for the case when the system (3) is not normalized (then X^a, Y^a, and U^a are unit matrices) are also calculated (A3). Let us consider, for example relative relationship between degrees of controllability [5] of the second modal variable by controls u_i, $i = 1,2,\ldots,8$. The relationship is given in Fig. 2a where the elements of Q_c are used and in Fig. 2b where the elements

	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8
z_2	1	4	8	3	6	2	5	7

Fig. 2a

	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8
z_2	4	1	3	6	8	2	7	5

Fig. 2b

of Q_c^a are used. The integers in the tables are associated to the degrees of controllability in such a manner that 1 corresponds to the maximal and 8 to the minimal absolute value of the corresponding matrix elements. From Fig. 2a it is evident that the influence of the throttle valve opening on the "fast" modal variable z_2 is the most significant. This can be expected from physical considerations of the plant properties. However, Fig. 2b suggests that the modal variable z_2 is the most controllable by the fuel input (which is in this case represented by the coal mill feeder stroke) which should not be expected. This example illustrates how the lack of the proper system normalization can lead to the wrong conclusions about the system structural properties.

CONCLUSION

Industrial process control system design requires analysis of the plant control properties based on the physical considerations and plant mathematical model. The plant mathematical model is usually nonlinear and it should be linearized to make it feasible for application of the linear system analysis methods. In this case, it is important to normalize the linear system variables. The normalization does not affect the modal analysis results, but it is essential for evaluation of the system structural properties. Candidate values for the proper normalization are given (or defined)maximal available perturbations of the system variables in the range of the plant operation under consideration, named the nominal perturbations. As was illustrated by an example, lack of the proper system variables normalization can lead to the wrong conclusions about the system structural properties.

REFERENCES

1. M. Healey, D. J. Mackinnon, "A Quantitative Measure of Observability for a Linear System," Int. J. Control, Vol. 22, No. 3, 1975.
2. P. J. Clelland, Power Plant Data Base Philadalphia Electric Company Cromby No. 2 Unit, ERDA 76-150/1, 1976.
3. H. G. Kwatny, J. P. McDonald, J. H. Spare, "A Nonlinear Model for Reheat Boiler-Turbine-Generator Systems, Part II - Development," Proc. 12th Joint Auto. Contr. Conf., 1971.
4. J. P. McDonald, "Computer-Aided Design and Analysis of Multivariable Process Controls with Application to a Boiler-Turbine-Generator System," Ph.D. thesis, Drexel University, 1972.
5. J. H. Chow, "Pole Placement Design of Multiple Controller Systems via Weak and Strong Controllability," Proc. 14th Annual Allerton Conf. on Circuit and System Theory, Univ. of Illinois, 1976.

Appendix A1

Table 1

Out. Var.	Process Variable	Limits (range)	y_b^o	y_n
y_1	generat.	200-220	210	20
y_2	reh.out. temp.	1461.12 ± 15°R	1461.	30
y_3	aver.reh. temperat.	1369.7 ± 15°R	1369.7	30
y_4	throttle temp.	1461.8 ± 15°R	1461.8	30
y_5	aver.sec. sup.temp.	1240.6 ± 15°R	1240.6	30
y_6	throttle pressure	1823.8 ± 50	1823.8	100
y_7	drum press.	1961.7 ± 50	1961.7	100
y_8	th.fl.-fw. fl-c d.l.	± 0.74	0.0	1.48

Table 2

Con var	Process variable	Limits (range)	u_b^o	u_n
u_1	gov. val. area	76.4% ± 12.5%	76.4%	25%
u_2	mill feed. stroke	66.2% ± 5%	66.2%	10%
u_3	reh.att. spr. flow	0-0.0167	0.0	0.0167
u_4	air flow	485.83 ± 60	485.83	60.
u_5	reh.bur. tilt pos.	± 30°	5.65	60.
u_6	sup.att. spr.flow	0-0.0167	0.0	0.0167
u_7	sup.bur. tilt pos.	± 30°	-19	60
u_8	feedwater val.area	74% ± 5%	74%	10%

Table 3

Sta. var.	Process variable	x_b^o	x_n
x_1	reh.out. enthalpy	1524	19.2
x_2	aver.reh. enthalpy	1421	17.88
x_3	sec.sup. out.ent.	1481.6	18.
x_4	aver.sec. sup.enth.	1402.4	17.
x_5	prim.sup. enthalpy	1323.2	16
x_6	reh.steam density	0.442	0.02
x_7	sec.sup. st. dens.	2.29	0.14
x_8	prim.sup. st.dens.	3.177	0.20
x_9	drum.st. density	5.2	0.33
x_{10}	dr.water volume	535	136.
x_{11}	sup.fur. met. temp.	1225	12.2
x_{12}	reh.fur. met.temp.	1223	12.2
x_{13}	fr.tot.m occ.by coal	0.585 $\times 10^{-3}$	0.58 x10
x_{14}	mass of coal in cr.zone	3705	37.0

APPENDIX A2

A

-3.10^{-3}	2.10^{-3}	-4.10^{-4}	0	0	-5.10^{-5}	-5.10^{-4}	0	0	0	0	$1.3.10^{-5}$	$4.7.10^{-5}$	0
3.10^{-4}	-4.10^{-3}	1.10^{-3}	0	0	7.10^{-4}	-2.10^{-3}	0	0	0	0	$1.4.10^{-5}$	5.10^{-5}	0
0	0	-2.10^{-3}	-4.10^{-3}	-4.10^{-3}	0	5.10^{-3}	-7.10^{-3}	0	0	$1.5.10^{-5}$	0	$3.5.10^{-5}$	0
0	0	6.10^{-3}	-5.10^{-3}	-4.10^{-3}	0	1.10^{-2}	-1.10^{-2}	0	0	$1.6.10^{-5}$	0	$3.7.10^{-5}$	0
3.10^{-5}	1.15^{-3}	6.10^{-5}	8.10^{-4}	2.10^{-2}	6.10^{-5}	1.10^{-4}	3.10^{-2}	-3.10^{-2}	0	1.10^{-6}	-2.10^{-5}	3.10^{-5}	0
-7.10^{-2}	0	0.1	0	0	-0.3	0.3	0	0	0	0	0	0	0
0	0	-0.4	0	0.5	0	-1	0.9	0	0	0	0	0	0
0	0	0.3	0	-1	0	0.7	-2	0.8	0	0	0	0	0
0	0	0	0	0.5	0	0	0.8	-0.8	0	0.1	0.1	0	0
0	0	0	0	7.10^{-2}	0	0	0.1	-0.1	0	5.10^{-3}	4.10^{-3}	0	0
0	0	0	0	0	0	0	0	0.01	0	-2.10^{-2}	0	$1.47.10^{-3}$	0
0	0	0	0	0	0	0	0	0.01	0	0	-2.10^{-2}	$1.43.10^{-3}$	0
0	0	0	0	0	0	0	0	0	0	0	0	-5.10^{-2}	5.10^{-2}
0	0	0	0	0	0	0	0	0	0	0	0	0	-1.10^{-2}

C

0.38	0	0.24	0	0	0.36	0.32	0	0	0	0	0	0	0
1.2	0	0	0	0	0.045	0	0	0	0	0	0	0	0
0.025	1.15	0.0015	0	0	0.06	0.0043	0	0	0	0	0	0	0
0	0	1.1	0	0	0	0.28	0	0	0	0	0	0	0
0	0	0.06	0.85	0.08	0	0.14	0.135	0	0	0	0	0	0
0	0	0.55	0	0	0	1.24	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0.91	0	0	0	0	0
0	0	0.006	0	0	0	0.018	0	-0.017	-1.14	0	0	0	0

B

1	2	3	4	5	6	7	8
$-3.8.10^{-4}$	$3.9.10^{-4}$	$-4.2.10^{-4}$	$2.4.10^{-3}$	$3.1.10^{-3}$	0	0	0
$2.8.10^{-3}$	$4.2.10^{-4}$	$-7.5.10^{-3}$	$2.6.10^{-3}$	$3.3.10^{-3}$	0	0	0
$-4.8.10^{-4}$	$2.9.10^{-4}$	0	$2.8.10^{-3}$	0	$-4.2.10^{-4}$	$3.5.10^{-3}$	0
0	3.10^{-4}	0	3.10^{-3}	0	$-9.5.10^{-3}$	$3.6.10^{-3}$	0
$3.4.10^{-6}$	$2.6.10^{-4}$	$3.9.10^{-6}$	$5.8.10^{-3}$	$-5.8.10^{-4}$	0	$3.1.15^{-4}$	0
0.24	0	0.27	0	0	0	0	0
$-6.4.10^{-2}$	0	0	0	0	$5.6.10^{-2}$	0	0
0	0	0	0	0	0	0	0
0	0	$-3.2.10^{-3}$	0	0	$-3.2.10^{-3}$	0	$5.7.10^{-3}$
0	0	$-3.7.10^{-3}$	0	0	$-3.7.10^{-3}$	0	$6.6.10^{-3}$
0	$1.23.10^{-2}$	0	$-1.33.10^{-2}$	0	0	$-8.5.10^{-3}$	0
0	$1.2.10^{-2}$	0	$-1.32.10^{-2}$	$-8.2.10^{-3}$	0	0	0
0	-0.4	0	0	0	0	0	0
0	0.15	0	0	0	0	0	0

D

-0.28	0	$-1.2.10^{-2}$	0	0	0	0	0
0	0	0	0	0	0	0	0
3.10^{-3}	0	$3.5.10^{-3}$	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
$1.3.10^{-2}$	0	$1.9.10^{-2}$	0	0	$1.1.10^{-2}$	0	-2.10^{-2}

APPENDIX A3

EIGENVALUES

$$
\begin{array}{cccccccccccccc}
\lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 & \lambda_5 & \lambda_6 & \lambda_7 & \lambda_8 & \lambda_9 & \lambda_{10} & \lambda_{11} & \lambda_{12} & \lambda_{13} & \lambda_{14} \\
0 & -2.5 & -0.9 & -0.3 & -0.09 & -0.02 & -0.007 & -0.004 & -0.004 & -0.0015 & -0.003 & -0.003 & -0.05 & -0.01 \\
 & & & & & & & +j0.002 & -j0.002 & & +j0.0003 & -j0.0003 & &
\end{array}
$$

$$
Q_0
$$

$$
\begin{bmatrix}
0 & 0.125 & 0.08 & 0.36 & -0.36 & -0.058 & -0.21 & 0.25 & -0.045 & 0.008 & 0.043 & -0.03 & -0.2 & -0.12 \\
0 & -0.003 & -0.018 & 0.045 & -0.029 & -0.006 & 0.095 & 0.22 & 0.11 & -0.067 & 0.2 & 0.16 & -0.017 & -0.015 \\
0 & -0.002 & -0.02 & 0.058 & -0.038 & -0.007 & -0.22 & 0.13 & -0.17 & -0.05 & 0.038 & -0.04 & -0.023 & -0.014 \\
0 & 0.13 & 0.2 & -1.2\cdot 10^{-5} & -0.12 & -0.023 & 0.12 & -0.16 & -0.03 & -0.033 & 0.017 & -0.03 & -0.08 & -0.06 \\
0 & -0.037 & 0.1 & 1.1\cdot 10^{-6} & -0.12 & -0.022 & -0.28 & 0.047 & 0.033 & -0.02 & 0.019 & -0.02 & -0.08 & -0.038 \\
0 & 0.6 & 0.9 & -5.5\cdot 10^{-5} & -0.54 & -0.1 & -0.67 & 0.82 & -0.056 & 0.06 & 0.093 & -0.015 & -0.34 & -0.2 \\
0 & 0.34 & -0.5 & -6.1\cdot 10^{-5} & -0.45 & -0.1 & -0.057 & 0.09 & -0.098 & -0.05 & 0.09 & -0.078 & -0.33 & -0.24 \\
-1.1 & -0.24 & 0.35 & 3.7\cdot 10^{-5} & 0.027 & 0.18 & 0.53 & -0.4 & 1 & 1.13 & -1.1 & 1.1 & 0.2 & 1
\end{bmatrix}
$$

$$
Q_0^a
$$

$$
\begin{bmatrix}
0 & 2.1 & -71 & 257 & -176 & -26 & -102 & 37 & -48 & 16 & -28 & -54 & -95 & -52 \\
0 & -3.4 & -19 & 47 & -29 & -4.1 & -13 & 0.6 & -8.5 & 2.3 & -5.5 & -9.7 & -15 & -8.3 \\
0 & -4 & -24 & 62 & -38 & -5.4 & -19 & 1.3 & -12 & 3.2 & -7.4 & -13 & -20 & -11 \\
0 & 28 & 40 & -0.003 & -25 & -4.7 & -38 & 46 & -2.2 & 3.9 & 4.4 & -0.07 & -16 & -8.8 \\
0 & -1.4 & 22 & 0.0002 & -21 & -4 & -25 & 20 & -3.5 & 3.1 & 3.5 & -0.75 & -14 & -7.2 \\
0 & 406 & 595 & -0.04 & -370 & -69 & -559 & 690 & -32 & 58 & 65 & -0.37 & -234 & -129 \\
-0.01 & 0.06 & 0.167 & -3\cdot 10^{-6} & -0.04 & -0.004 & -0.1 & 0.13 & -0.013 & 0.028 & -0.0055 & 0.02 & -0.02 & 0.003
\end{bmatrix}
$$

$$Q_c$$

$$\begin{bmatrix}
0.017 & -1.5 & 0.015 & 0.83 & 0.27 & 0.02 & 0.26 & 1.3\cdot 10^{-6} \\
-0.026 & -0.0035 & -0.0013 & -0.009 & 0.0025 & 0.02 & 0.003 & 0.0023 \\
-0.054 & 0.018 & 0.0026 & -0.01 & -0.0057 & 0.05 & 0.0006 & -0.0045 \\
0.24 & 0.054 & 0.27 & -0.05 & -0.01 & 0.04 & -0.015 & -0.0044 \\
0.03 & 0.38 & 0.002 & -0.2 & -0.056 & -0.023 & -0.065 & -0.004 \\
1.1\cdot 10^{-5} & 0.0008 & 5.7\cdot 10^{-7} & 8.4\cdot 10^{-5} & -0.006 & -9\cdot 10^{-6} & 0.006 & -1.8\cdot 10^{-6} \\
-0.004 & 0.16 & 0.007 & -0.016 & -0.012 & 0.02 & -0.013 & 0.0001 \\
0.0005 & 0.09 & -0.0001 & 0.01 & -0.004 & 0.01 & -0.01 & 1.9\cdot 10^{-5} \\
-0.024 & 0.48 & 0.045 & -0.17 & -0.078 & -0.03 & -0.046 & 0.00095 \\
-0.032 & 0.9 & 0.056 & -0.58 & -0.17 & 0.06 & -0.18 & 0.0028 \\
-0.038 & 1.85 & 0.03 & -0.8 & -0.24 & 0.02 & -0.24 & 0.0042 \\
0.048 & -2.3 & -0.06 & 0.85 & 0.3 & -0.003 & 0.26 & -0.0046 \\
0 & -0.78 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1.4 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

$$Q_c^a$$

$$\begin{bmatrix}
-0.062 & -657 & -13.5 & 0.165 & 0.055 & 9.7 & 0.05 & 9 \\
-0.012 & -0.43 & -0.05 & -0.0026 & 0.0006 & -0.23 & 0.001 & 0.0076 \\
-0.024 & 2.1 & 0.058 & -0.0013 & -0.0012 & -0.3 & 0.0008 & -0.015 \\
-0.007 & 5.8 & -0.04 & -0.01 & -0.0018 & -0.16 & -0.0028 & -0.014 \\
0.013 & 36 & -0.18 & -0.04 & -0.01 & -0.32 & -0.014 & -0.013 \\
5\cdot 10^{-6} & 0.2 & -0.0004 & 2\cdot 10^{-5} & -0.0012 & -0.0006 & 0.0013 & -6\cdot 10^{-6} \\
-0.055 & 65 & 8 & -0.0027 & -0.003 & 22 & -0.003 & 0.00035 \\
0.02 & 35 & -0.2 & 0.003 & -0.001 & 11 & -0.0031 & -6\cdot 10^{-5} \\
-0.3 & 205 & 50 & -0.04 & -0.018 & -41 & -0.008 & 0.003 \\
-0.35 & 390 & 63 & -0.13 & -0.036 & 58 & -0.04 & 0.009 \\
-0.28 & 785 & 38 & -0.17 & -0.05 & 2.9 & -0.049 & 0.014 \\
0.36 & -979 & -62 & 0.174 & 0.07 & 20 & 0.053 & -0.015 \\
0 & -94 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 509 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

GRADIENT ESTIMATION ALGORITHMS FOR ADAPTIVE EQUALIZATION IN THE DISCRETE FREQUENCY DOMAIN

PAOLO CORSINI*, GIORGIO PICCHI* and GIANCARLO PRATI**
*Dipartimento Sperimentale di Elettrotecnica ed Elettronica, Università di Pisa, Pisa, Italy.
**Centro di Studio per i Metodi e i Dispositivi di Radiotrasmissione, C.N.R., Pisa, Italy.

ABSTRACT. An adaptive version of the linear non-recursive automatic equalizer in the discrete frequency domain for synchronous data transmission is considered. Algorithms for estimating the gradient of the mean-square error in the discrete frequency domain and for deriving the amplitude of the correction step are developed in detail for the "overlap-save" filtering method.

INTRODUCTION

Much attention has been devoted in the recent past to the problem of implementing a transversal discrete filter by using FFT algorithms [1], according to either the "overlap-save" method or the "overlap-add" one. The advantages consist essentially in achieving less hardware (or software) complexity, if the number of taps is suitably large.

On the other hand a transversal filter with adjustable tap gains is often used in equalizing a discrete channel for synchronous data transmission. Usually the equalizing filter is implemented in the time domain and is automatically adjusted by means of an equalizer controller either in a time period preceding the actual data transmission (preset equalizer) or during the same (adaptive equalizer).

What follows from the above is the opportunity of examining the behavior of an equalizer entirely implemented in the discrete frequency domain, i.e., where the filter is designed by using the above recalled FFT techniques and the controller computes and modifies a vector $\mathbf{C}$ representing the DFT of a suitably expanded sequence of the filter tap weights.

Walzman and Schwartz [2] proposed a preset equalizer in the discrete frequency domain, using isolated test pulses as training signals. They found that their equalizer showed a better speed of convergence with respect to the corresponding one in the time domain proposed by Gersho [3].

In this paper we consider an adaptive equalizer in the discrete frequency domain employing the overlap-save method for data filtering. The overlap-save method has been preferred rather than the overlap-add one, because the output data samples depend in the first method, only on the present filter configuration, while in the second they are obtained by using also quantities computed with the previous different filter configuration. Then, even if the filter configuration changes during the transmission, in the first method the filtering is always correct, while in the second it is (slightly?) incorrect.

In our study we adopt the mean square error as a cost function and the equalizer controller acts on the vector $\mathbf{C}$ according to an iterative procedure strongly related to the classical iterative gradient search using the Rosen's gradient projection method [2].

Algorithms to perform the previous operation are given in the following. They permit to estimate the gradient of the mean-square error and the amplitude of the correction step.

DFD-EQUALIZER STRUCTURE

Consider a discrete channel for synchronous data transmission and assume that its impulse response is finite and equal to $x_0, x_1, \ldots, x_{M-1}$. Moreover, assume that the transmitted sequence

$$a_0, a_1, \ldots \tag{1}$$

is a sample sequence of a discrete ergodic process with uncorrelated quantized symbols. This channel can be adaptively equalized by using a M-tap transversal filter in the time domain, whose tap weights are automatically adjusted by a suitable equalizer controller. In this paper we consider the corresponding adaptive equalizer having the transversal filter implemented via FFT techniques using the overlap-save method.

The overall structure of the equalizer in discrete frequency domain (called from now on DFD-equalizer) is shown in Fig. 1.

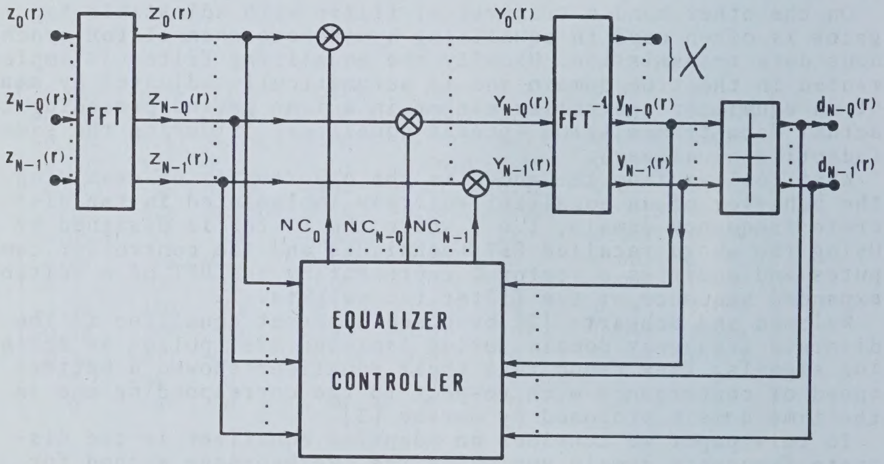


Fig.1-Overall structure of the equalizer.

The equalizing filter acts on the channel output sequence

$$z_0, z_1, \ldots \tag{2}$$

and produces as output the sequence

$$y_0, y_1, \ldots \tag{3}$$

The corresponding quantized sequence is

$d_0, d_1, \ldots$ (4)

At the r-th filtering operation the equalizer processes the vector†

$$\mathbf{z}(r) = \left(z_0(r) \ z_1(r) \ \ldots \ z_{N-1}(r)\right)^T$$

where $z_n(r) = z_{rQ+n}$, $n=0,1,\ldots,N-1$, so that two consecutive vectors $\mathbf{z}(r-1)$ and $\mathbf{z}(r)$ are overlapped of N-Q terms. The vector $\mathbf{Z}(r)$ is the DFT of the vector $\mathbf{z}(r)$, i.e.

$$\mathbf{Z}(r) = \frac{1}{N} W \mathbf{z}(r)$$

where W is a NxN matrix whose (i,k)-th element is $\exp(-j2\pi ik/N)$.

The equalizer controller produces, apart from a multiplicative constant N, the parameter vector

$$\mathbf{C} = \left(C_0 \ C_1 \ \ldots \ C_{N-1}\right)^T$$

which must represent the DFT of a vector $\left(c_0 \ c_1 \ \ldots \ c_{M-1} \ 0 \ \ldots \ 0\right)^T$ obtained by appending N-M zeros to the M tap coefficients $c_0, c_1, \ldots, c_{M-1}$ of the corresponding transversal filter in the time domain. Then, the vector $\mathbf{C}$ must satisfy to the constraints

$\bar{W}\mathbf{C} = W\bar{\mathbf{C}}$ (reality of the DFT^{-1} of $\mathbf{C}$) (5a)

$\mathbf{C} = P\mathbf{C}$ (N-M zeros as last elements of the DFT^{-1} of $\mathbf{C}$) (5b)

where $P = W \, Diag\left(\underbrace{1 \ \ldots \ 1}_{M} \ \underbrace{0 \ \ldots \ 0}_{N-M}\right) W^{-1}$

The vector at the input of the FFT^{-1} structure can be written in matrix form as

$\mathbf{Y}(r) = N \, Diag \mathbf{Z}(r) \, \mathbf{C}$ (6)

Consider the vector $\mathbf{y}(r) = \bar{W}\mathbf{Y}(r)$ representing the DFT^{-1} of the vector $\mathbf{Y}(r)$. Owing to the circularity error, only the last Q terms of $\mathbf{y}(r)$, $Q \quad N-M+1$, can be assumed as correct filtered samples. These terms are $y_{N-Q}(r)$, $y_{N-Q+1}(r), \ldots, y_{N-1}(r)$ and represent the elements y_{rQ+N-Q}, $y_{rQ+N-Q+1}, \ldots, y_{rQ+N-1}$ of the sequence (3). The corresponding quantized output samples $d_{N-Q}(r)$, $d_{N-Q+1}(r), \ldots, d_{N-1}(r)$ are the elements d_{rQ+N-Q}, $d_{rQ+N-Q+1}, \ldots$, d_{rQ+N-1} of the sequence (4).

---

† Lower case letters will denote time domain quantities and capitals will denote frequency domain quantities. Boldface letters denote column vectors. Capitals can also denote matrices and integers numbers. The transponse of a vector $\mathbf{V}$ (matrix B) is $\mathbf{V}^T$ (B^T) and its complex conjugate is $\bar{\mathbf{V}}$ ($\bar{B}$).

THE MEAN-SQUARE ERROR

In this study we adopt the mean square error as a cost function and we will adjust the parameter vector $\mathbf{C}$ to minimize it.

Due to the fact that the equalizer parameters are changed during the data transmission, the output time-discrete error process y_i-a_i is not stationary, even if the input process a_i is ergodic and the channel is time-invariant. Then, the mean-square error $<(y_i-a_i)>$ at the DFD-equalizer output depends on i, that is on the time. On the other hand the mean-square error, for a particular choice of $\mathbf{C}$, can be considered a random variable (function of $\mathbf{C}$) having the expected value equal to the mean-square error of the non-stationary process at the same time [4, pag. 208]. As a consequence we can study the problem as in a stationary case, so that the conditioned mean-square error

$$\varepsilon = <(y_i-a_i)^2|\mathbf{C}> \tag{7}$$

does not depend on i. Then, by assuming that the output quantized samples (4) are correct, we can write the error on a sample y_i as $e_i=y_i-d_i$. For a particular choice of $\mathbf{C}$ (which satisfies to the constraints (5)),the output mean-square error ε of the time-discrete output process is†

$$\varepsilon = <e_i^2> = <(y_i-d_i)^2> \tag{8}$$

and it does not depend on i. Then, we have also

$$\varepsilon = \frac{1}{Q}\sum_{j=N-Q}^{N-1} <(y_j(r)-d_j(r))^2>$$

Introducing $\mathbf{Y}'(r)$ and $\mathbf{D}'(r)$ as the DFTs of the expanded vectors $\mathbf{y}'(r)=(0 \ \dots \ 0 \ y_{N-Q}(r) \ \dots \ y_{N-1}(r))^T$ and $\mathbf{d}'(r)=(0 \ \dots \ 0 \ d_{N-Q}(r) \ \dots \ d_{N-1}(r))^T$ and using the Parseval relationship we have

$$\varepsilon = \frac{N}{Q} < ||\mathbf{Y}'(r)-\mathbf{D}'(r)||^2>$$

Noting that $\mathbf{Y}'(r)=P'\mathbf{Y}(r)$ where P' is the projection matrix $W\, Diag(\underbrace{0 \ \dots \ 0}_{N-Q} \ \underbrace{1 \ \dots \ 1}_{Q})\, W^{-1}$ and using (6), we can write

$$\varepsilon = \mathbf{C}^T A\bar{\mathbf{C}} - [\mathbf{B}^T\bar{\mathbf{C}}+\bar{\mathbf{B}}^T\mathbf{C}] + \frac{N}{Q}<||\mathbf{D}'(r)||^2> \tag{9}$$

where

$$A = \frac{N^3}{Q}<[P' \, Diag\,\mathbf{Z}(r)]^T \ [\bar{P}' \, Diag\,\bar{\mathbf{Z}}(r)]>$$

$$\mathbf{B}^T = \frac{N^2}{Q} < \mathbf{D}'(r)^T \ Diag\,\bar{\mathbf{Z}}(r)>$$

and $\mathbf{C}$ is chosen consistent with the constraints (5).

---

† From now on the symbol $< \cdot >$ is to be intended as $< \cdot |\mathbf{C}>$

In the next section, eq. (9) is used to derive an algorithm according to which the equalizer controller modifies the vector $\mathbf{C}$. The algorithm uses the quantity

$$\nabla\varepsilon=\left(\frac{\partial\varepsilon}{\partial Re(C_0)}+j\,\frac{\partial\varepsilon}{\partial Im(C_0)}\quad\cdots\quad\frac{\partial\varepsilon}{\partial Re(C_{N-1})}+j\,\frac{\partial\varepsilon}{\partial Im(C_{N-1})}\right)^T \tag{10}$$

Applying definition (10) to (9), we obtain

$$\nabla\varepsilon=\frac{2N^2}{Q}\ \langle Diag\,\bar{\mathbf{Z}}(r)\,\mathbf{E}(r)\rangle \tag{11}$$

where $\mathbf{E}(r)=\mathbf{Y}'(r)-\mathbf{D}'(r)$.

EQUALIZER CONTROLLER ALGORITHMS

The starting point to derive the algorithms implemented in the equalizer controller is the following statement:

The minimization of the mean-square error ε under the constraints (5) can be performed by modifying the vector $\mathbf{C}$ according to the iterative procedure

$$\mathbf{C}^{(k+1)}=\mathbf{C}^{(k)}-\frac{1}{2}\,\alpha\,PG\,\nabla^{(k)}\varepsilon \qquad k=0,1,\ldots \tag{12}$$

where α is a suitable constant and

$$G=Diag\left(\underbrace{\frac{1}{2}\ 1\ \ldots\ 1\ \frac{1}{2}}_{(N/2)+1}\ \underbrace{1\ \ldots\ 1}_{(N/2)-1}\right)$$

Any solution of the constraints (5) may be used as starting point $\mathbf{C}^{(0)}$.

A complete proof of the statement is omitted. Briefly, the main steps for the proof are:

(i) ε can be written as a real function of N independent real variables (this possibility is due to constraint (5a)) in the form

$$\varepsilon=\mathbf{C}_w^T A_r \mathbf{C}_w-2\mathbf{B}_r^T\mathbf{C}_w+\frac{N}{Q}\ \langle\,\|\mathbf{D}'(r)\|^2\rangle \tag{13}$$

where

$$\mathbf{C}_w=\left(Re(C_0)\ \ldots\ Re(C_{N/2})\ \ Im(C_1)\ \ldots\ Im(C_{N/2\ -1})\right)^T$$

and $A_r,\mathbf{B}_r$ are proper rearrangments of A and $\mathbf{B}$;

(ii) the constraint (5b) implies

$$\mathbf{C}_w=P_w\mathbf{C}_w \tag{14}$$

P_w being the projection operator (16) in [2];

(iii) minimization of ε in (13) under the constraint (14) can be performed according to the iterative gradient-search using the Rosen's gradient projection method

$$\mathbf{C}_w^{(k+1)}=\mathbf{C}_w^{(k)}-\frac{1}{2}\,\alpha\,P_w\,\nabla_w^{(k)}\varepsilon \tag{15}$$

where

$\nabla_w^{(k)}\varepsilon=2A_r\mathbf{C}_w^{(k)}-2\mathbf{B}_r$

Expression (15) can be written in complex terms as in (12).

Implementation of eq. (12) requires the knowledge of the quantities $\nabla^{(k)}\varepsilon$ and α . Matrix PG has to be computed only once off-line, since it is not function of the input samples. The problem of finding a suitable value of α is encountered in the next section.

An unbiased estimate $\hat{\nabla}^{(k)}\varepsilon$ of $\nabla^{(k)}\varepsilon$ in (11) may be computed by the equalizer controller on the basis of R output blocks of Q samples as follows

$$\hat{\nabla}^{(k)}\varepsilon=\frac{2N^2}{RQ}\sum_{r=kR}^{(k+1)R-1} Diag\,\bar{\mathbf{Z}}(r)\,\mathbf{E}(r) \qquad k=0,1,... \qquad (16)$$

DERIVING AN ESTIMATE OF α

The starting point to derive an estimate for the quantity α to be used in eq. (12) is the following statement:

An optimum value of α in the sense of minimizing an upper bound for the norm of the vector $\mathbf{C}^{(k)}-\mathbf{C}_{opt}$ is

$$\alpha=\frac{2}{\rho_{max}+\rho_{min}} \qquad (17)$$

where ρ_{max} and ρ_{min} are respectively the maximum and minimum eigenvalue of the diagonal matrix

$$A_w=2N^3G\,Diag\mathbf{X}\;Diag\bar{\mathbf{X}} \qquad (18)$$

$\mathbf{X}$ being the DFT of $\mathbf{x}=(x_0\ ...\ x_{M-1}\ 0\ ...\ 0)^T$.

A complete proof of the statement is omitted. Briefly the main steps for the proof are:

(i) the mean-square error ε_w introduced by Walzman and Schwartz in [2] for the case of isolated test pulses coincides, apart from a multiplicative constant (the average power S of the transmitted sequence (1)),with ε of eq. (13), if the constraint (14) is taken into account. This means that ε_w and ε represent in general two different hypersuperficies in the same (N+1)-dimensional space defined by the vector $\mathbf{C}_w$, which have essentially the same intersection with the hyperplane (14);

(ii) the gradient of ε_w and ε are in general different, but their projections into the manifold Γ defined by (14) are the same;

(iii) as a consequence of (ii), the optimum value of α (in the sense of maximizing the speed of convergence of ε) must be the same in the present case and in [2], so that the statement holds.

Since α in (17) depends on $\mathbf{X}$, an estimate $\hat{\mathbf{X}}$ of $\mathbf{X}$ must be obtained.

We have found that an unbiased estimate on the basis of RQ received samples is

$$\hat{\mathbf{X}}=P\,\frac{N}{RQS}\sum_{r=0}^{R-1} \mathit{Diag}\,\mathbf{Z}(r)\,\bar{\mathbf{D}}(r) \tag{19}$$

where $\mathbf{D}(r)$ is the DFT of the vector $\left(d_0(r)\ \dots\ d_{Q-1}(r)\ 0\ \dots\ 0\right)^T$

To justify informally eq. (19), observe that x_m can be obtained in the time domain by cross-correlating sequences (4) and (2). Then an unbiased estimate $\hat{x}_m$ of x_m on the basis of RQ received samples in the time domain is

$$\hat{x}_m=\frac{1}{RQS}\sum_{u=0}^{RQ-1} d_u z_{m+u} \qquad m=0,1,\dots,M-1$$

By utilizing the properties of the DFT of correlated sequences, we obtain that the vector

$$\hat{\mathbf{X}}'=\frac{N}{RQS}\sum_{r=0}^{R-1} \mathit{Diag}\,\mathbf{Z}(r)\,\bar{\mathbf{D}}(r)$$

is the DFT of a vector having $\hat{x}_m$, $m=0,1,\dots,M-1$, as first elements and not significative quantities as last elements. Then $P\hat{\mathbf{X}}'$ is the estimate $\hat{\mathbf{X}}$ of $\mathbf{X}$.

CONCLUDING REMARKS

Equations (12), (16), (17) and (19) define completely a controller for an equalizer in the discrete frequency domain using the overlap-save method for filtering operations.

This equalizer has been simulated for the same channel as in [2] and the results are summarized in Figs. 2,3 (curve 1). In Fig. 2 the estimates are obtained with M=16, N=32, Q=16, R=100, so that the vector $\mathbf{C}$ is modified every 1600 received samples. In Fig. 3 the estimates are obtained with R=1, i.e. every 16 samples, which corresponds to the maximum adjustment rate for the proposed equalizer.

It should be noted that in the above equalizer the estimate (16) is made using only the data samples received after the previous estimate has been completed. A proper utilization of $\hat{\mathbf{X}}$ (see eq. (18) in [2]), however, permits to find a gradient projection of ε_w, which we are allowed to use (see step (ii) in the previous section) instead of $P\nabla_w^{(k)}\varepsilon$ in eq. (15). This alternative way of estimating the projected gradient does not suffer from the above limitation, because each estimate $\hat{\mathbf{X}}$ does not depend on $\mathbf{C}^{(k)}$ and can use many preceding samples (if the channel is time-invariant or changes slowly) to improve the confidence. However, this algorithm shows a greater complexity than the previous one. Simulation results for the same channel are presented again in Fig. 2 (curve 2, M=16, N=32, Q=16, R=100) and in Fig. 3 (curve 2, M=16, N=32, Q=16, R=1). Figs. 2,3 may be interpreted taking into account the following information. Results of 25 sample runs were averaged to obtain plots. Vertical width of the shaded stripes represents standard deviation from the sample mean. Curve 3 represents the ε mean value in the limit case of exactly estimated gradient.

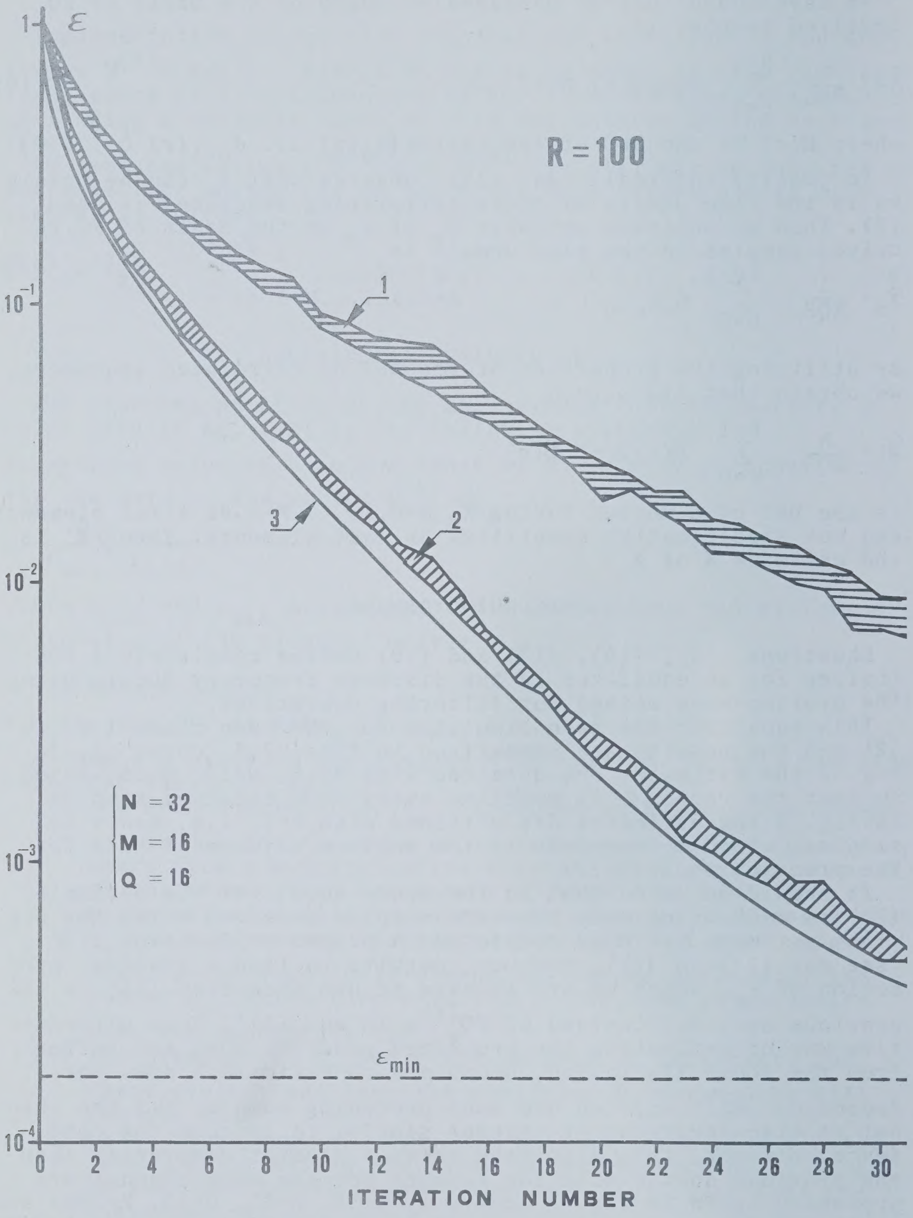


Fig.2-Simulation results: gradient estimation over 1600 samples.

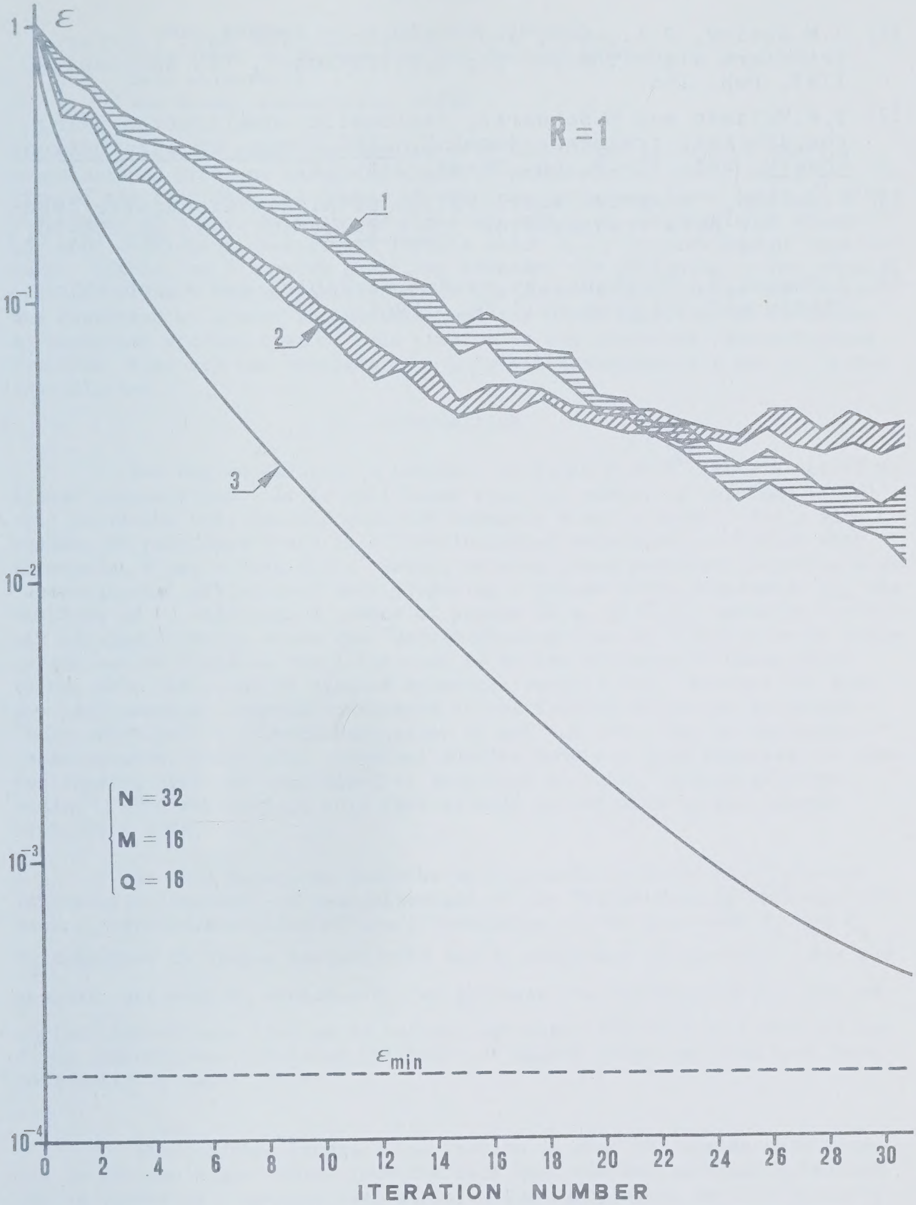


Fig.3-Simulation results: gradient estimation over 16 samples.

REFERENCES

[1] J.W.Cooley, P.A.Lewis, and P.D.Welch, "The fast Fourier transform algorithm and its applications", IBM Res.Rep.,RC 1743, Feb. 1967

[2] T.W.Walzman and M.Schwartz, "Automatic equalization using the discrete frequency domain", IEEE Trans. on Information Theory, vol. IT-19, pp. 59-68, Jan. 1973

[3] A.Gersho, "Adaptive equalization of highly dispersive channels for data transmission", B.S.T.J., vol. 48, pp. 55-70, Jan. 1969

[4] A.Papoulis, "Probability random variables and stochastic processes", Mc Graw Hill Co., 1965

ON A CLASS OF POLYTOPE INTERSECTION PROBLEMS

B. ROSS BARMISH and JAISHANKAR SANKARAN
Department of Engineering and Applied Science
Yale University
New Haven, Connecticut 06520

Abstract. In this paper, we develop some algorithms to solve a class of computational problems involving (convex) polytopes and their vertices. In view of the fact that the number of vertices increases "very rapidly" with the number of faces, we must pay close attention to computational feasibility when developing theoretical results which apply to problems of this nature. Within the framework above, we consider the following rather general problem: Find a vertex description of the intersection of two polytopes--one described by linear inequalities and the other described by vertices. An important special case of this problem is the classical Vertex-Finding Problem: Find all the vertices of a polytope described by a set of linear inequalities.

I. INTRODUCTION

One way to describe a (convex) polytope P in R^n is by a set of m linear inequalities. It is well-known that the number of vertices V(n,m) of P increases very rapidly with the integers m and n above. For a wide variety of polytopes there is a "combinatorial explosion" of V(n,m) with increasing m and n (see [1]). Hence, we must place paramount importance on "computational efficiency" when proposing a scheme which enumerates all the vertices of P. Although a number of papers (e.g. [2]-[5]) describe a variety of algorithms to solve the "Vertex-Finding Problem," little or no indication can be found in the literature as to the efficacy of these algorithms from the point of view of numerical computation. Perhaps the most popular numerical example considered in the literature is the so-called "baby polytope" of [4]--characterized by m=8 and n=3. To the knowledge of these authors, meaningful numerical results have not been reported for vertex-finding problems that might be construed as being "medium to large scale." (The enormity of this task is well illustrated by the example $V(10,20) \leq 4004!$).

In this paper, we describe an algorithm to solve the "Polytope Intersection Problem"--a generalization of the Vertex-Finding Problem. We seek a vertex description of the intersection of two polytopes P_I and P_v - P_I described by linear inequalities and P_v described by vertices. For the special case when P_v contains P_I, we generate the vertices of P_I. Our numerical experiments lead us to believe our algorithm will be practical for doing computations involving polyhedra of higher dimension than have been considered to date.

The Polytope Intersection Problem is also of considerable interest in its own right (aside from the fact that the Vertex-Finding Problem can be viewed as a special case of it). For example, in certain problems of state estimation, the set of possible states of a dynamical system can be described by the intersection of two polytopes--one polytope arising from parameter uncertainty in the so-called state equation and the other polytope arising from parameter uncertainty in the so-called observation equation (as in [8]).

II. FORMULATION

The given data consists of two polytopes P_I and P_v. The inequality polytope P_I is described by

$$P_I = \{x \varepsilon R^n : \langle a_i, x \rangle \leq b_i;\ i = 1,2,\ldots,m\}$$

where $a_i \varepsilon R^n$, $b_i \varepsilon R$ are given and $\langle \cdot, \cdot \rangle$ denotes the usual inner product; P_v is a vertex polytope described by

$$P_v = \mathrm{conv}\{v_1, v_2, \ldots, v_p\}$$

where $v_i \varepsilon R^n$ are the given vertices (extreme points) of P_v and $\mathrm{conv}\{v_1, v_2, \ldots, v_p\}$ denotes the convex hull of the v_i. To avoid trivialities, we assume that all of the a_i are non-zero and the v_i are distinct.

2.1 Polytope Intersection Problem (PIP):

To find a vertex description of the polytope

$$P = P_I \cap P_v.$$

We note that when P_v contains P_I, we are solving the

2.2 Vertex-Finding Problem (VFP):

To find the vertices of P_I given the vertices of a bounding polytope $P_B = P_v$ which contains P_I.

III. CONCEPTUAL APPROACH - WITHOUT COMPUTATIONAL STREAMLINING

In this section we discuss the solution of Problem (PIP) conceptually. However, to make this scheme computationally tractable, we have later in this paper included certain "streamlining" techniques which will substantially reduce computation.

Unlike the (VFP) algorithms of [4] and [5], our approach does not involve manipulation of the simplex tableau. It is closer in spirit to the (VFP) algorithm in [6] in that we add one inequality at a time. In contrast to [6], we do not work with cones and their extreme rays. Instead, we work with a certain set-valued mapping Λ which transforms P into $\Lambda(P)$, a new polytope whose vertices are easier to compute than those of P. Furthermore, we give a counterexample to show that the "adjacency criterion" in [6] is in fact not sufficient for deciding whether or not a given candidate point $v \varepsilon R^n$ is an extreme point of P. In light of this counterexample, we develop a certain rank test which can be used in testing for extremality.

To describe our approach to finding the vertices of P, we need some notation. We introduce the half spaces $H_i^- = \{x \varepsilon R^n : \langle a_i, x \rangle \leq b_i\}$; $i = 1,2,\ldots,m$; and hyperplanes $H_i = \{x \varepsilon R^n : \langle a_i, x \rangle = b_i\}$; $i = 1,2,\ldots,m$. Now, we define the sequence of polytopes $P_0 = P_v$, $P_1 = P_0 \cap H_1^-$, $P_2 = P_1 \cap H_2^-$, $\ldots, P_m = P_{m-1} \cap H_m^-$. Then it follows easily that

$$P_m = P = P_I \cap P_v. \qquad (3.1)$$

Consequently, the solution of (PIP) hinges upon the answer to the following question: Knowing the vertices of P_{k-1}, can we update this information to P_k? (We note that the recursion can be initialized because the vertices of $P_0=P_v$ are known.) This leads us to consider

3.1 Subproblem $(PIP)_k$:

Let $v_1(k-1), v_2(k-1), \ldots, v_{r(k-1)}(k-1)$ denote the vertices of P_{k-1}. Find the vertices $v_1(k), v_2(k), \ldots, v_{r(k)}(k)$ of P_k.

3.2 Remark:

In the analogous subproblem in [6], a double description of P_{k-1} is required. i.e., it is assumed that a linear inequality description of P_{k-1} is available in addition to its vertices. Here, we dispense of this a priori assumption. We show that a double description of P_{k-1} is not essential to the development of the theory--but having this additional information turns out to be advantageous in the context of numerical computation (see Section 4).

We now describe a sequence $\Lambda_0, \Lambda_1, \Lambda_2, \ldots, \Lambda_m$ of set-valued mappings. These maps will be generated recursively and the map Λ will be defined by $\Lambda = \Lambda_m$.

For $x \varepsilon R^n$, we define

$$\Lambda_k(x) = \{\lambda \varepsilon S^k : \sum_{i=1}^{r(k)} \lambda^i v_i(k) = x\}; \ k = 0,1,2,\ldots,m$$

where S^k is the unit simplex in $R^{r(k)}$. i.e.,

$$S^k = \{\lambda \varepsilon R^{r(k)} : \lambda \geq 0; \sum_{i=1}^{r(k)} \lambda^i = 1\}.$$

Some properties of this map are immediate.

Property 1: $\Lambda_k(x) \neq \phi$ if and only if $x \varepsilon P_k$.

Property 2: If x is an extreme point of P_k, say $x = v_j(k)$, then the set $\Lambda_k(x)$ consists of a single point. i.e., $\Lambda_k(v_j(k)) = \{e_j\}$ where e_j is a unit vector in the j^{th} coordinate direction.

Property 3: If $x \neq y$, then $\Lambda_k(x) \cap \Lambda_k(y) = \phi$.

Property 4: $\Lambda_k^{-1}(S^k) = P_k$.

3.3 Remark:

We can think of $\Lambda_k(x)$ as the collection of coefficients (generally non-unique) which can be used to generate x as a convex combination of the $v_i(k)$. The following lemma is an easy consequence of (3.1), Properties 2 and 4, and the fact that $\Lambda = \Lambda_m$.

3.4 Lemma: $\Lambda^{-1}(S^m) = P$. *Moreover, the set* extP, *of extreme points of* P, *is given by*

$$\text{ext}P = \{\Lambda^{-1}(e_1), \Lambda^{-1}(e_2), \ldots, \Lambda^{-1}(e_{r(m)})\}.$$

As mentioned earlier, we are going to develop an updating procedure to go from P_{k-1} to P_k. It will be easier (conceptually and computationally) to first develop a characterization of the "polytope" $\Lambda_{k-1}(P_k)$ rather than P_k itself. Once this has been accomplished, we will describe an algorithm to generate P_k from $\Lambda_{k-1}(P_k)$. The following lemmas facilitate the computation of $\Lambda_{k-1}(P_k)$.

3.5 Lemma: $\Lambda_{k-1}(P_k) = \Lambda_{k-1}(H_k^-)$.

Proof: $\Lambda_{k-1}(P_k) = \Lambda_{k-1}(H_k^- \cap P_{k-1}) = \Lambda_{k-1}(H_k^-) \cap \Lambda_{k-1}(P_{k-1})$

$= \Lambda_{k-1}(H_k^-) \cap S^{k-1} = \Lambda_{k-1}(H_k^-)$. ☐

3.6 Lemma: *Suppose* $\lambda \varepsilon R^{r(k-1)}$. *Then* $\lambda \varepsilon \Lambda_{k-1}(P_k)$ *if and only if*

$$\lambda \varepsilon S^{k-1}; \tag{3.2}$$

$$\sum_{i=1}^{r(k-1)} \lambda^i \langle a_k, v_i(k-1)\rangle \le b_k. \tag{3.3}$$

Proof: (Necessity) Suppose $\lambda \varepsilon \Lambda_{k-1}(P_k)$. Then by definition of Λ_{k-1}, $\lambda \varepsilon S^{k-1}$ and moreover $x = \Sigma_{i=1}^{r(k-1)} \lambda^i v_i(k-1) \varepsilon P_k$. By the previous lemma, it follows that $x \varepsilon H_k^-$. Hence,

$$b_k \ge \langle a_k, x\rangle = \langle a_k, \sum_{i=1}^{r(k-1)} \lambda^i v_i(k-1)\rangle = \sum_{i=1}^{r(k-1)} \lambda^i \langle a_k, v_i(k-1)\rangle.$$

(Sufficiency) Suppose λ satisfies (3.2) and (3.3). Then the vector $x = \Sigma_{i=1}^{r(k-1)} \lambda^i v_i(k-1)$ satisfies $\langle a_k, x\rangle \le b_k$. Hence $x \varepsilon P_{k-1} \cap H_k^- = P_k$. Since $x \varepsilon P_k$ and $\lambda \varepsilon S^{k-1}$, it follows that $\lambda \varepsilon \Lambda_{k-1}(P_k)$. ☐

3.7 Computation of $\Lambda_{k-1}(P_k)$:

We make use of (3.2) and (3.3) by first noting that there are three possibilities for the set

$$H_\lambda^-(k) = \{\lambda \varepsilon R^{r(k-1)}: (3.3) \text{ is satisfied}\}.$$

Case 1: $\langle a_k, v_i(k-1)\rangle \ne 0$ for some $i \le r(k-1)$. Then $H_\lambda^-(k)$ is a half-space corresponding to the hyperplane $H_\lambda(k)$ which is described by

$$H_\lambda(k) = \{\lambda: \sum_{i=1}^{r(k-1)} \lambda^i \langle a_k, v_i(k-1)\rangle = b_k\}$$

Case 2: $\langle a_k, v_i(k-1)\rangle = 0$ for all $i \le r(k-1)$; $b_k \ge 0$. Then $H_\lambda^-(k) = R^{r(k-1)}$.

Case 3: $\langle a_k, v_i(k-1)\rangle = 0$ for all $i \le r(k-1)$; $b_k < 0$. Then $H_\lambda^-(k) = \phi$.

We can easily dispose of Cases 2 and 3 because

$$\Lambda_{k-1}(P_k) = S^{k-1} \text{ for Case 2}$$

and

$$\Lambda_{k-1}(P_k) = \phi \text{ for Case 3.}$$

Henceforth, we assume that $\langle a_k, v_i(k-1)\rangle \ne 0$ for some $i \le r(k-1)$ and we are seeking an extreme point description of the set

$$\Lambda_{k-1}(P_k) = S^{k-1} \cap H_\lambda^-(k).$$

3.7.1 Remark: Loosely speaking, the extreme points of $\Lambda_{k-1}(P_k)$ will be a subset of the set points where $H_\lambda(k)$ meets the "edges" of S^{k-1} plus those unit vectors lying in $H_\lambda^-(k)$.

3.7.2 Extreme Points of $\Lambda_{k-1}(P_k)$: To make the preceding ideas precise, we define edge (i,j) of S^{k-1} as

$$E_{ij} = \{\alpha e_i + (1-\alpha)e_j: \alpha \varepsilon (0,1)\};\ i \ne j;\ i,j = 1,2,\dots,r(k-1).$$

In order for $H_\lambda(k)$ to meet E_{ij}, it is necessary and sufficient to have $\alpha_* \varepsilon (0,1)$ such that $\lambda_* = \alpha_* e_i + (1-\alpha_*)e_j$ satisfies

$$\sum_{i=1}^{r(k-1)} \lambda_*^i \langle a_k, v_i(k-1)\rangle = b_k.$$

There are two cases to consider

Case 1: $\langle a_k, v_i(k-1)\rangle \ne \langle a_k, v_j(k-1)\rangle$. Then $E_{ij} \cap H_\lambda(k) \ne \phi$ if and only if there exists α_* such that

$$\alpha_* = \frac{b_k - \langle a_k, v_j(k-1)\rangle}{\langle a_k, v_i(k-1)\rangle - \langle a_k, v_j(k-1)\rangle}\ ; \tag{3.4}$$

$$0 < \alpha_* < 1. \tag{3.5}$$

Case 2: $\langle a_k, v_i(k-1)\rangle = \langle a_k, v_j(k-1)\rangle$. Then,

$$E_{ij} \cap H_\lambda(k) = E_{ij} \text{ if } \langle a_k, v_i(k-1)\rangle = b_k\ ;$$

$$= \phi \text{ if } \langle a_k, v_i(k-1)\rangle \ne b_k.$$

The following proposition summarizes our findings.

3.7.3 Proposition: The extreme points of $\Lambda_{k-1}(P_k)$ consist of those unit vectors e_i such that $<a_k,v_i(k-1)> \leq b_k$ plus all vectors of the form $\alpha_* e_i + (1-\alpha_*)e_j$ where α_* satisfies (3.4) and (3.5).

3.8 Computation of P_k from $\Lambda_{k-1}(P_k)$:

To complete our updating procedure, we must find the extreme points of P_k given the set (computed using Proposition 3.7.3)

$$\text{ext}\Lambda_{k-1}(P_k) \triangleq \{\lambda_1(k),\lambda_2(k),\ldots,\lambda_{q(k)}(k)\};\ \lambda_i(k)\varepsilon S^{k-1}.$$

This can be accomplished with the aid of some definitions and lemmas. For each $\lambda_i(k)$ above, we define

$$g_i(k) = \sum_{j=1}^{r(k-1)} \lambda_i^j(k)v_j(k-1);\ i = 1,2,\ldots,q(k). \qquad (3.6)$$

We call the set of $g_i(k)$ generating points (or g-points) for P_k. We shall see that the set of extreme points of P_k is in general a (proper) subset of the set of g-points of P_k. The name "generating points" is made more meaningful by the following lemma.

3.8.1 Lemma: $P_k = \text{conv}\{g_1(k),g_2(k),\ldots,g_{q(k)}(k)\}$.

Proof: Suppose $x\varepsilon P_k$. Then $x\varepsilon P_{k-1}$ and $<a_k,x> \leq b_k$. We write x as

$$x = \sum_{i=1}^{r(k-1)} \beta^i v_i(k-1);\ \beta\varepsilon\Lambda_k(P_{k-1}). \qquad (3.7)$$

Now, we express β as a convex combination of the $\lambda_i(k)$. i.e.,

$$\beta^i = \sum_{j=1}^{q(k)} \alpha^j\lambda_j^i(k);\ i = 1,2,\ldots,r(k-1)$$

where

$$\sum_{j=1}^{q(k)} \alpha^j = 1;\ \alpha^j \geq 0;\ j = 1,2,\ldots,q(k).$$

Substituting for β in (3.7), we have

$$x = \sum_{i=1}^{r(k-1)} \sum_{j=1}^{q(k)} \alpha^j\lambda_j^i(k)v_i(k-1) = \sum_{j=1}^{q(k)} \sum_{i=1}^{r(k-1)} \alpha^j\lambda_j^i(k)v_i(k-1) = \sum_{j=1}^{q(k)} \alpha^j g_j(k).$$

Hence, $x\varepsilon\text{conv}\{g_1(k),g_2(k),\ldots,g_{q(k)}(k)\}$ and we have $P_k \subseteq \text{conv}\{g_1(k),g_2(k),\ldots,g_{q(k)}(k)\}$.

To show the reverse containment, suppose x is a convex combination of the $g_i(k)$. i.e.,

$$x = \sum_{i=1}^{q(k)} \alpha^i g_i(k); \ \sum_{i=1}^{q(k)} \alpha^i = 1; \ \alpha^i \geq 0; \ i = 1,2,\ldots,q(k).$$

Then, arguing as above, we obtain

$$x = \sum_{i=1}^{q(k)} \beta^i v_i(k-1); \ \sum_{i=1}^{q(k)} \beta^i = 1; \ \beta^i \geq 0; \ i = 1,2,\ldots,q(k).$$

Hence, $x \varepsilon P_{k-1}$. Furthermore,

$$\langle a_k, x\rangle = \langle a_k, \sum_{j=1}^{q(k)} \sum_{i=1}^{r(k-1)} \alpha^j \lambda_j^i(k) v_i(k-1)\rangle = \sum_{j=1}^{q(k)} \alpha^j \sum_{i=1}^{r(k-1)} \lambda_j^i(k) \langle a_k, v_i(k-1)\rangle$$

$$\leq \sum_{j=1}^{q(k)} \alpha^j b_k \qquad (\text{since } \lambda_j(k) \varepsilon \Lambda_{k-1}(P_k), \text{ see } (3.3)) \qquad = b_k.$$

So we also have $x \varepsilon H_k^-$. Hence, $x \varepsilon P_{k-1} \cap H_k^- = P_k$ and we have $\mathrm{conv}\{g_1(k), g_2(k), \ldots, g_{q(k)}(k)\} \subseteq P_k$. □

3.8.2 Lemma (see [9], Corollary 5): *A system* $\langle a_i, x\rangle < 0$; $i = 1, 2, \ldots, m$ *of strict homogeneous linear inequalities in* R^n *is consistent if and only if* $0 \varepsilon \mathrm{conv}\{a_1, a_2, \ldots, a_m\}$.

As a fairly straightforward consequence of the above lemma, we have the following result which enables us to test for the extremality of $g_\ell(k)$ even though a double description of P_v is not available a priori. The proof of the following lemma can be found in [12].

3.8.3 Lemma: *A g-point* $g_\ell(k)$ *is extreme if and only if*

$$\{x: \langle g_j(k) - g_\ell(k), x\rangle \leq -\varepsilon; \ j = 1,2,\ldots,q(k); \ j \neq \ell\} = \phi; \ \forall \varepsilon > 0. \qquad (3.8)$$

3.8.4 Remark: Lemma 3.8.3 points to a computational procedure for deciding which of the g-points are in fact extreme points of P_k. We simply pick some "small" $\varepsilon > 0$ and compute a *new* (perhaps empty) set of g-points for the polytope $\tilde{P}_\varepsilon = \tilde{P}_{I,\varepsilon} \cap \tilde{P}_v$ where $\tilde{P}_{I,\varepsilon}$ is the inequality polytope described by the left hand side of (3.8) and $\tilde{P}_v$ is the vertex polytope given by $\tilde{P}_v = \{x \varepsilon R^n: |x^i| \leq 1; \ i = 1,2,\ldots,n\}$. We note that a double description of $\tilde{P}_v$ is readily available. Consequently, the so-called "efficiency theory" of Section 4 can be used in deciding whether or not $\tilde{P}_\varepsilon$ is empty. Finally, we conclude that $g_\ell(k)$ is extreme if $\tilde{P}_\varepsilon = \phi$; $g_\ell(k)$ is not extreme if $\tilde{P}_\varepsilon \neq \phi$. Note that this method enables us to solve the following general problem--given $x_0, x_1, \ldots, x_p \varepsilon R^n$, does $x_0 \varepsilon \mathrm{conv}\{x_1 \ldots x_p\}$?

3.9 Summary of Conceptual Approach:

At this point, we can summarize our approach to updating the set of extreme points from P_{k-1} to P_k. In the next section, we develop some theory which enables us to streamline this procedure from the point of view of numerical computation.

Given: We have $v_i(k-1) \varepsilon \text{ext} P_{k-1}$; $i = 1,2,\ldots,r(k-1)$.

Step 1: Throw in H_k^-. Use equations (3.2), (3.3) to obtain the inequality description of the polytope $\Lambda_{k-1}(P_k)$.

Step 2: Use Proposition 3.7.3 to obtain $\lambda_i(k) \varepsilon \text{ext} \Lambda_{k-1}(P_k)$; $i = 1,2,\ldots,r(k)$.

Step 3: Use equation (3.6) to obtain $g_i(k)$; $i = 1,2,\ldots,q(k)$.

Step 4: Use Section 3.8.4 to obtain $v_i(k) \varepsilon \text{ext} P_k$.

IV. EFFICIENCY THEORY

Our objectives in this section are twofold:

(1) We wish to describe some simple innovations which will substantially reduce the computational effort associated with Steps 2 and 4 above. Consequently, we shall replace Steps 2 and 4 by Steps 2* and 4* respectively.

(2) Under the strengthened hypotheses that are present in the Vertex-Finding Problem (VFP), we show that Step 4 (the most time-consuming computational step) can be replaced by a simplified "rank test" or "label check" which can be used to decide which g-points are extreme.

4.1 Description of Steps 2* and 4*:

We recall that Step 2 involves finding the intersection of $H_\lambda(k)$ and E_{ij}. In practice, this would involve checking conditions (3.4) and (3.5) for all possible edges E_{ij}. We reduce computational effort by exploiting the following basic fact: $\Lambda_{k-1}(P_k)$ has an extreme point in E_{ij} only if e_i and e_j lie on opposite sides of $H_\lambda(k)$. Hence,

Step 2* consists of

(1) Checking $<a_k, v_i(k-1)> \leq b_k$

(2) Checking conditions (3.4) and (3.5) for the subset of edges described above.

The simplification of Step 4 is accomplished with the aid of another basic fact: If $g_\ell(k)$ is a g-point of P_k, which is an extreme point of P_{k-1}, then $g_\ell(k)$ is an extreme point of P_k. Hence,

Step 4* consists of checking the condition "$\tilde{P}_\varepsilon = \phi$" only for a subset of the g-points of P_k--those g-points of P_k which are not extreme points of P_{k-1}.

4.2 The Vertex-Finding Problem:

For this important special case of (PIP) we can develop some theory which is useful in further reducing computation associated with Step 4. Without loss of generality, we henceforth assume that the bounding polytope P_B is an n-rectangle described by

$$P_B = \{x \varepsilon R^n: x_-^i \leq x^i \leq x_+^i;\ i = 1,2,\ldots,n\}$$

where x_-^i, x_+^i are scalars which are known a priori. (Note: If P_B is not given in this form, we simply compute the smallest rectangle containing P_B and use that instead.) Since we can assume that P_B is rectangular, a double description of P_B is readily available.

4.2.1 Basic Question: Under the strengthened hypothesis on $P_v=P_B$ for the (VFP), can we give necessary and sufficient conditions under which a g-point is extreme? (Of course, we are seeking conditions which require less computational effort than required by the condition "$\tilde{P}_\varepsilon = \phi$.")

4.2.2 Adjacency Criterion: In [6], a certain adjacency criterion is suggested in an attempt to answer the basic question above. In this section, we describe this condition and give a counterexample in Section 4.2.3, which shows that this condition is not sufficient for extremality. Let $v_i(k-1)$ and $v_j(k-1)$ be extreme points of P_{k-1}. Then, we call $v_i(k-1)$ and $v_j(k-1)$ adjacent if

A1. $v_i(k-1)$, $v_j(k-1)$ lie in (n-1) or more common hyperplanes $\tilde{H}_1,\tilde{H}_2,\ldots,\tilde{H}_r$.

A2. No other extreme point v of P_k lies in $\tilde{H}_1,\tilde{H}_2,\ldots,\tilde{H}_r$.

We may conjecture that any g-point of P_k arising via (3.6) from some $\lambda_i(k)\varepsilon E_{ij}$ will be extreme. This conjecture turns out to be false as seen below.

4.2.3 Counterexample: We consider the polytope P_{k-1} in R^6 described by

$$|x_i| \leq 5,\ i = 1,5;\ x_i \leq 0,\ i = 2,3,4;\ -x_i \leq 5,\ i = 3,4;$$

$$-x_6 \leq 10;\ -x_2 + x_3 + x_4 \leq 0;\ x_1 + x_5 + x_6 \leq 0.$$

It can easily be shown that the pair of points $v_i(k-1) = (5,0,0,0,5,-10)$ and $v_j(k-1) = (-5,0,0,0,-5,10)$ are adjacent. Now, if H_k^- is described by $-x_1 - x_5 \leq 5$, upon application of the algorithm, we obtain a g-point $g_\ell(k) = (-5/2,0,0,0,-5/2,5)$ which is not an extreme point of P_k because $g_\ell(k)$ can be expressed as $g_\ell(k) = \frac{1}{2}v_1(k) + \frac{1}{2}v_2(k)$; $v_1(k),v_2(k)\varepsilon P_k$ where $v_1(k) = (-5,0,0,0,0,5)$ and $v_2(k) = (0,0,0,0,-5,5)$.

4.2.4 Conditions Under Which a g-Point is Extreme:

Preliminary Notation: Given a point $x\varepsilon R^n$, at stage k, we use the matrix $A_k(x)$ and the vector $b_k(x)$ to describe the collection of those inequalities encountered as equality at x so far. (By "so far" we mean the inequalities describing P_0 and $<a_i,x> \leq b_i$; i = 1,2,...,k.) In other words, $A_k(x)x = b_k(x)$.

From the theory of linear programming (c.f. [12], p. 19), we know that if $x_*\varepsilon R^n$, then

$$x_*\varepsilon \text{ext} P_k \text{ if and only if } \operatorname{rank} A_k(x_*) = n. \tag{4.1}$$

Of course, we have in mind the case where x_* is a g-point of P_k. If we further take into account the mechanism by which g-points are generated, we get the following stronger result.

4.2.5 Theorem: *Suppose $g_\ell(k)$ is a g-point of P_k which is not an extreme point of P_{k-1}. Then*

$$g_\ell(k) \varepsilon \text{ext} P_k \text{ if and only if } \operatorname{rank} A_{k-1}(g_\ell(k)) = n - 1. \qquad (4.2)$$

(i.e., we can neglect the inequality $<a_k,x> \leq b_k$ in testing for extremality.) We need one fundamental lemma in order to prove the theorem.

4.2.6 Lemma: *Suppose $g_\ell(k)$ is a g-point of P_k such that* rank $A_{k-1}(g_\ell(k)) = n - 1$. *Then* $\Lambda_{k-1}(g_\ell(k))$ consists of a single point. (Note: this lemma will hold if $g_\ell(k)$ is replaced by an arbitrary vector $x \varepsilon P_k$.)

4.2.7 Remark: The proof of this theorem and lemma, being quite lengthy, has been omitted here for brevity. The proof is available in its entirety in [12].

4.2.8 Further Reduction in Computation: In the previous section, we concluded that a new g-point $g_\ell(k)$ is extreme if and only if $A_{k-1}(g_\ell(k))$ has rank n-1. To check this condition, we would have to compute det $A'_{k-1}(g_\ell(k))A_{k-1}(g_\ell(k))$ or carry out the Gram-Schmidt procedure (c.f. [11], pp. 54-55). These potentially expensive calculations can be reduced with the aid of the labelling algorithm described below.

4.2.9 Labelling Algorithm: To describe the algorithm we require a definition.

Definition: Suppose $g_\ell(k)$ is a g-point of P_k. We define the carriers of $g_\ell(k)$ to be that pair of vertices $v_i(k)$, $v_j(k)$ such that $g_\ell(k) = \lambda^i v_i(k) + \lambda^j v_j(k)$ for some $\lambda^i, \lambda^j \geq 0$, $\lambda^i + \lambda^j = 1$. (Note: If $g_\ell(k) \varepsilon \text{ext} P_{k-1}$, then we must take $v_i(k-1) = v_j(k-1) = g_\ell(k)$. i.e. $g_\ell(k)$ is its own carrier.)

Step 0 (Initialization): Attach the label u to all vertices of the bounding polytope P_B.

Step k: Suppose all vertices of P_{k-1} are labelled and we wish to update our labels to P_k. We label g-points of P_k as follows:

Case 1: $g_\ell(k) \varepsilon \text{ext} P_{k-1}$

If $g_\ell(k)$ carries c, label remains as c; if $g_\ell(k)$ carries u, label remains as u if $g_\ell(k) \notin H_k$, label becomes c if $g_\ell(k) \varepsilon H_k$.

Case 2: $g_\ell(k) \notin \text{ext} P_{k-1}$

Label u if one (or both) carriers is labelled u; label c otherwise.

Theorem 4.2.5 tells us that only some of the g-points must be tested for extremality--those g-points $g_\ell(k)$ such that $A_{k-1}(g_\ell(k))$ has at least n-1 rows.

The following theorem (a strengthened version of Theorem 4.2.5) tells us that we can further reduce the number of g-points which must undergo a rank test. (Our early computational experience for a wide variety of examples indicates that the set of g-points requiring a rank test is in some sense "scarce.")

4.2.10 Theorem (see [12] for proof): Suppose $g_\ell(k)$ is a g-point of P_k which is not an extreme point of P_{k-1}. Then $g_\ell(k) \varepsilon \mathrm{ext} P_k$ if and only if one or more of the following two conditions hold:

(i) $g_\ell(k)$ carries the label u and $A_{k-1}(g_\ell(k))$ has at least n-1 rows.

(ii) rank $A_{k-1}(g_\ell(k)) = n - 1$.

4.2.11 Summary of Streamlined Algorithm for (VFP):

Step 0 (Initialization): Attach the label u to all vertices of the bounding polytope $P_0=P_B$, set k=1.

Step 1: Suppose $v_1(k-1), v_2(k-1), \ldots, v_{r(k-1)}(k-1)$ are labelled vertices of P_{k-1}. Compute, using (3.4), (3.5) and Proposition 3.7.3 the set $\mathrm{ext}\Lambda_{k-1}(P_k) = \{\lambda_1(k), \lambda_2(k), \ldots, \lambda_{q(k)}(k)\}$.

Step 2: Compute, using (3.6), the g-points $g_\ell(k), g_2(k), \ldots, g_{q(k)}(k)$ of P_k.

Step 3: Label the g-points above in accordance with Section 4.2.9.

Step 4: Declare as "extreme" those g-points of P_k which belong to $\mathrm{ext}P_{k-1}$.

Step 5: From the remaining set of g-points, declare as "extreme" those points which carry the label u.

Step 6: From the remaining set of g-points, declare as "extreme" those g-points $g_\ell(k)$ such that rank $A_{k-1}(g_\ell(k)) = n - 1$.

Step 7: Reject the remaining g-points. We are left with vertices $v_1(k), v_2(k), \ldots, v_{r(k)}(k)$, those g-points which have been declared extreme.

Step 8: If k=m, terminate; otherwise, replace k by k+1 and return to Step 1. (When the algorithm terminates, $v_1(m), v_2(m), \ldots, v_{r(m)}(m)$ will be the vertices of the inequality polytope P_I.)

4.2.12 Remark: The general streamlined algorithm to solve (PIP) and (VFP) has been implemented in APL on the IBM 370 - Model 158. Our numerical results are reported in [12] where we were able to compute vertices for problems as large as (m,n) = (14,7) with less than 20 sec. of c.p.u. time.

REFERENCES

[1] V. Klee, "On the Number of Vertices of a Convex Polytope," Canadian Jour. Math., vol. 16, (1964), pp 701-20.
[2] D. Gale, "On the Number of Faces of a Convex Polytope," Canadian Jour. Math., vol. 16, (1964), pp 12-17.
[3] M. Fieldhouse, "Linear Programming," Ph.D. Thesis, Cambridge Univ., 1961 (Reviewed Jour. Operations Research, vol. 10, (1962), p 740).

[4] M. L. Balinski, "An Algorithm for Finding all Vertices of a Convex Polyhedral Set," J. Siam, vol. 9, (1961), pp 72-88.

[5] T. H. Mattheiss, "An Algorithm for Determining Irrelevant Constraints and Vertices in Systems of Linear Inequalities," J. Op. Research, vol. 21, (1973), pp 247-60.

[6] T. S. Motzkin, H. Raiffa, G. Thompson and R. Thrall, "The Double Description Method," Contributions to the Theory of Games, vol. 2, H. W. Kuhn and A. W. Tucker (eds.), Annals of Math. Studies, (1953), pp 51-73.

[7] M. Manas and J. Nedoma, "Finding all Vertices of a Convex Polyhedron," Numerische Mathematik, vol. 12, (1968), pp 226-29.

[8] J. Sankaran and B. R. Barmish, "State Estimation in the Presence of Uncertain Dynamics," Proc. 1977 Joint Automatic Control Conf., pp 1089-1094.

[9] K. Fan, "On Systems of Linear Inequalities," Contributions to the Theory of Games, vol. 2, H. W. Kuhn and A. W. Tucker (eds.), Annals of Math. Studies, (1953), pp 99-156.

[10] R. T. Rockafellar, Convex Analysis, Princeton Univ. Press, 2nd ptg., 1972.

[11] D. G. Luenberger, Optimization by Vector Space Methods, Wiley, NY, 1969.

[12] B. R. Barmish, J. Sankaran and H. Tachibana, "On Intersecting Polytopes - Theory and Computation," S & IS Report No. 7707, Department of Engineering and Applied Science, Yale University, (1977).

SPACE-TIME TRADEOFFS ON THE FFT ALGORITHM* (Extended Abstract)

JOHN E. SAVAGE and SOWMITRI SWAMY
Program in Computer Science, Brown University, Providence, RI 02912

ABSTRACT

The performance of the Fast Fourier Transform Algorithm is examined under limitations on computational space and time. It is shown that if the algorithm with n inputs, n a power of two, is implemented with S temporary storage locations where $S = o(n/\log n)$, then the computation time T grows faster than $n \log n$. Furthermore, T can grow as fast as n^2 if $S = S_{min} + O(1)$, where $S_{min} = 1 + \log_2 n$, the minimum storage necessary. These results are obtained by deriving tight bounds on T versus S and n.

INTRODUCTION

The Fast Fourier Transform (FFT) algorithm enjoys a special place in the history of computer science because it is one of the earliest demonstrations of the power of algorithmic analysis. The FFT is an algorithm for computing the discrete Fourier transform (or its extensions) on n inputs in $O(n \log n)$ steps as opposed to $O(n^2)$ steps for the standard algorithm. The FFT is widely used in signal processing.

The FFT has been independently discovered several times and the paper by Cooley and Tukey [1] has lead to its widespread use. An early history of the algorithm is reported in [2].

The FFT algorithm is basically a straight-line algorithm in which the operations are executed in a data-independent manner. When the algorithm is used on a machine with a limited number of temporary storage locations, however, it is necessary to repeat many of the sub-computations. This leads to a storage-time tradeoff, the topic of this paper. The tradeoff problem for the FFT algorithm has interest because this algorithm is often used in a limited storage space environment and because it is an example of a "natural" problem for which the storage-time tradeoff problem can be studied.

The storage-time tradeoff problem for a straight-line algorithm can be modeled by the "pebble game" on a directed graph. The nodes of such a graph correspond to operations, and edges denote dependencies between operations. In the pebble game, pebbles are moved from input to output nodes of the graph subject to three conditions: (1) a pebble can be placed on an input node at any time, (2) a pebble can be removed from a node at any time, and (3) a pebble can be placed on a non-input node only when all nodes with edges directed into this node have pebbles. Pebbles correspond to temporary registers in a machine and the maximum number of pebbles used, S, is a measure of storage space. Time T is measured as the number of moves necessary to pebble the graph when at most S pebbles are allowed.

The pebble game was introduced by Paterson and Hewitt [3] and studied by several other authors. Hopcroft, Paul, and Valiant [4] have shown that a graph on N nodes can be pebbled using space $O(N/\log N)$, where the constant of proportionality depends on the maximum in-degree of nodes. Paul, Tarjan, and Celoni [5] have also shown the existence of arbitrarily large graphs for which every pebbling strategy uses space $\Omega(N/\log N)$. Concerning the exchange of space for time, Pippenger [6] has exhibited a graph on N nodes for which $2T/N = \log_2 N/2S + O(1)$ so that more than linear time (in N) is necessary to pebble the graph with space $O(N/\log N)$. Chandra [7] has examined the result of Paterson and Hewitt [3], which can be expressed as a space-time tradeoff, and has shown that more than linear time is required if $\log_2 S = o(\log_2 N)$. And finally Paul and Tarjan [8] have shown the existence graphs on N nodes such that reduction in S by a constant factor causes T to expand from $O(N)$ to $2^{O(\sqrt{N})}$.

In this paper we obtain fairly tight bounds on the time required to pebble a graph which is a simple homomorphic image of the graph of the FFT algorithm on $n = 2^d$ inputs. The exchange of time for space which is derived in following sections can be stated in the memorable but weaker form:

$$n^2 \frac{11}{128} - \frac{n \log_2 n}{8} \leq (S - \log_2 n)\ (T - n \log_2 n) \leq 4n^2$$

Thus, the locus of space-time exchanges lies between two hyperbola in the space-time plane. If $S = o(n/\log n)$, T grows more rapidly than $O(n \log n)$ and approaches $O(n^2)$ as S approaches its minimum value.

The key to the derivation of the bounds on time is a representation of an FFT graph as the composition of sets of smaller FFT graphs. This result is developed in Section 2. The following sections present a pebbling strategy, the derivation of lower bounds and conclusions.

THE FFT ALGORITHM

A recursive derivation of the Fast Fourier Transform (FFT) is given by Borodin and Munro [9] and we assume that the reader is familiar with this derivation. The FFT algorithm is a straight-line algorithm and can be described by a directed acyclic graph, as shown in Figure 1 when the number of inputs n = 8. The FFT graph in general is recursively defined, as given below.

Definition Let $n = 2^d$, $d \geq 0$. The directed graph $F^{(d)} = (V^{(d)}, D^{(d)}, E^{(d)})$ associated with the FFT algorithm on n-sequences has node set $V^{(d)}$, edge set $E^{(d)} \subseteq V^{(d)} \times V^{(d)}$ and output node set $D^{(d)} \subseteq V^{(d)}$ and it is defined recursively in terms of two disjoint FFT graphs ${}_0F^{(d-1)} = ({}_0V^{(d-1)}, {}_0D^{(d-1)}, {}_0E^{(d-1)})$ and ${}_1F^{(d-1)} = ({}_1V^{(d-1)}, {}_1D^{(d-1)}, {}_1E^{(d-1)})$ on (n/2)-sequences as follows: Let $D^{(d)} = \{v_0^{(d)}, v_1^{(d)}, \ldots v_{2^d-1}^{(d)}\}$ be a set of nodes that is disjoint from ${}_0V^{(d-1)}$ and ${}_1V^{(d-1)}$. Then,

$$V^{(d)} = {}_0V^{(d-1)} \cup {}_1V^{(d-1)} \cup D^{(d)}$$

$$E^{(d)} = {}_0E^{(d-1)} \cup {}_1E^{(d-1)} \cup \Delta^{(d)}$$

where

$$\Delta^{(d)} = \bigcup_{i=0}^{2^{d-1}} \delta_i^{(d)}$$

$$\delta_i^{(d)} = \{(v_i^{(d)}, {}_0v_i^{(d-1)}), (v_i^{(d)}, {}_1v_i^{(d-1)}), (v_{i+2^{d-1}}^{(d)}, {}_0v_i^{(d-1)}), (v_{i+2^{d-1}}^{(d)}, {}_1v_i^{(d-1)})\}$$

and (v, v') denotes an edge directed from v' to v. Also, $F^{(0)} = (V^{(0)}, V^{(0)}, \phi)$ where $V^{(0)} = \{v_0^{(0)}\}$ and ϕ is the empty set.

Consider the FFT graph in Figure 1 for n = 8. The construction of $F^{(3)}$ from ${}_0F^{(2)}$ and ${}_1F^{(2)}$ by the addition of output nodes and edges to them from output nodes of ${}_0F^{(2)}$ and ${}_1F^{(2)}$ should be clear. The graph has four levels with inputs at level o and outputs at level d = 3. There are four edge sets $\delta_i^{(3)}$, $0 \leq i \leq 3$, and each set with corresponding input and output nodes form an FFT graph on 2-sequences. In particular, that associated with $\delta_0^{(3)}$ is $(\{v_0^{(3)}, v_4^{(3)}, {}_0v_0^{(2)}, {}_1v_0^{(2)}\}, \{v_0^{(3)}, v_4^{(3)}\}, \delta_0^{(3)})$. These four FFT graphs are disjoint and take their inputs from the outputs of ${}_0F^{(2)}$ and ${}_1F^{(2)}$. A generalization of these observations is contained in the following lemma.

Lemma 1 For $d \geq 2$ and each $1 \leq j \leq d-1$ the FFT graph $F^{(d)}$ can be represented as the composition of 2^{d-j} disjoint FFT graphs **on** 2^j inputs $\{A_\ell^{(j)} | 0 \leq \ell \leq 2^{d-j}-1\}$ with 2^j disjoint FFT graphs on 2^{d-j} inputs $\{B_m^{(d-j)} | 0$

$\leq m \leq 2^j-1\}$ where the m^{th} input to $A_\ell^{(j)}$ is the ℓ^{th} output of $B_m^{(d-j)}$ for $0 \leq m \leq 2^j-1$ and $0 \leq \ell \leq 2^{d-j}-1$. The i^{th} output of $F^{(d)}$ is the r^{th} output of $A_s^{(j)}$ for the unique integers $r, s \geq 0$ such that $i = r2^{d-j} + s$ and $s < 2^{d-j}$.

Proof (By induction).

Basis $d = 2$, the hypothesis is true by inspection.

Inductive Step Assume that the hypothesis is true for $d \leq d_0-1$ and show that it holds for $d = d_0$. Let $j = 1$. Then $F^{(d_0)}$ is formed from ${}_0F^{(d_0-1)}$ and ${}_1F^{(d_0-1)}$ by adding the output nodes $D^{(d_0)}$ and edges $\Delta^{(d_0)}$. But the graphs $(\{v_i^{(d_0)}, {}_0v_i^{(d_0-1)}, v_{i+2^{d_0-1}}^{(d_0)}, {}_1v_i^{(d_0-1)}\}, \{v_i^{(d_0)}, v_{i+2^{d_0-1}}^{(d_0)}\}, \delta_i^{(d_0)})$, $0 \leq i \leq 2^{d_0-1}$, are 2^{d_0-j} FFT graphs $\{A_\ell^{(1)}\}$ on two inputs so the hypothesis follows from the definition of $F^{(d_0)}$ when $j = 1$.

Now let $2 \leq j \leq d_0-1$ and consider the graphs ${}_0A_\ell^{(j-1)}$, ${}_0B_m^{(d_0-j)}$ and ${}_1A_\ell^{(j-1)}$, ${}_1B_m^{(d_0-j)}$ into which ${}_0F^{(d_0-1)}$ and ${}_1F^{(d_0-1)}$ may be decomposed for $0 \leq \ell \leq 2^{d_0-j}-1$ and $0 \leq m \leq 2^j-1$. These four sets of graphs are disjoint except for nodes that the first two and last two sets have in common. The graphs ${}_0A_\ell^{(j-1)}$ and ${}_1A_\ell^{(j-1)}$ together with the output nodes $\{v_{i(s)}^{(d_0)}, v_{i(s)+2^{d_0-1}}^{(d_0)} \mid i(s) = \ell 2^{d_0-j} + s, 0 \leq s \leq 2^{d_0-1}-1\}$ form an FFT graph on 2^j inputs $A_\ell^{(j)}$, as can be seen from the definition of an FFT graph. Furthermore, there are 2^{d_0-j} such graphs $\{A_\ell^{(j)}\}$ where the m^{th} input to $A_\ell^{(j)}$ is the ℓ^{th} output of ${}_0B_m^{(d_0-j)}$ if $0 \leq m \leq 2^{d_0}-1$ or the ℓ^{th} output of ${}_1B_m^{(d_0-j)}$ if $2^{d_0-j} \leq m \leq 2^{d_0}-1$. If we let $B_m^{(d_0-j)}$ be ${}_0B_m^{(d_0-j)}$ in the first case and ${}_1B_m^{(d_0-j)}$ in the second, the conclusion follows.

This decomposition of an FFT graph is very useful in deriving upper and lower bounds to the time required to pebble the graph. Another fact is needed to derive lower bounds and this is contained in the next lemma.

Lemma 2 In $F^{(d)}$ the paths from any given input to outputs form a complete binary tree with 2^d leaves as do the paths to any given output from inputs.

Proof The hypothesis as it applies to paths to a given output follows directly from the definition of $F^{(d)}$ since there are exactly two edges directed into an output node from disjoint FFT subgraphs ${}_0F^{(d-1)}$ and ${}_1F^{(d-1)}$. The hypothesis on paths from a given input follows from Lemma 1 when $j=d-1$ since the two edges directed away from an input are directed to the two disjoint FFT subgraphs $A_0^{(d-1)}$ and $A_1^{(d-1)}$ of $F^{(d)}$.

We are now prepared to present a pebbling strategy for the FFT graph.

A PEBBLING STRATEGY

As stated in the introduction, a pebbling of a straight-line algorithm and its representing graph is a movement of a limited number of pebbles from input nodes to output nodes according to a few simple rules.

1) A pebble can be placed at any time on an input node.
2) A pebble can be removed at any time from any node.
3) A pebble can be placed on a node only when both of its immediate input nodes have pebbles on them.

The object of the pebbling game on a graph G is to move pebbles to the designated output nodes of G in a minimum number of moves, T, when the number of pebbles on the graph at any time is at most S.

The computational problem that is modeled by the pebbling game is the computation of a straight-line algorithm in which S measures the number of temporary registers that can be addressed and T is the number of cycles of computation. For example, the code for the straight-line algorithm could be read on-line, in which case S measures the total storage capacity used for data storage. Neither S nor T measures the space or time used for address calculation so they provide lower bounds on the total time and space needed. The pebbling is said to be done "without interpretation" because the computation sequence does not take into account the values of variables $a_o, a_1, \ldots, a_{n-1}$.

We now describe a pebbling strategy for $F^{(d)}$ and derive an upper bound on T in terms of S and d. Before we do, however, note that $F^{(d)}$ can be pebbled in $(d + 1)\ 2^d$ **moves** with $S = 2^d + 1$ pebbles by pebbling it in levels, starting with the input nodes at level 0. The number of moves on non-input nodes is $d2^d$. Also, it should be clear that a complete binary tree with k + 1 levels and 2^k leaves can be pebbled in $2^{k+1}-1$ moves with k + 1 pebbles.

To pebble $F^{(d)}$ with $S = 2^j + d - j$ pebbles, $1 \leq j \leq d-1$, represent $F^{(d)}$ as indicated in Lemma 1 and pebble $A_\ell^{(j)}$ using 2^j+1 pebbles. Pebble the inputs of $A_\ell^{(j)}$ by pebbling the 2^j subtrees of $B_m^{(d-j)}$, $0 \leq m \leq 2^j-1$, rooted on the ℓ^{th} outputs of each graph. Pebbles are left on each of the input nodes of $A_\ell^{(j)}$ so all of the S pebbles will be used. The number of moves made on $\{B_m^{(d-j)}\}$ to pebble the input nodes of $A_\ell^{(j)}$ is $2^j(2^{d-j+1}-1)$ since the subtrees of $\{B_m^{(d-j)}\}$ have depth d - j. An additional $j2^j$ moves are made on non-input nodes of $A_\ell^{(j)}$. Since this is done for each $0 \leq \ell \leq 2^{d-j}-1$, we have the following upper bound.

Theorem 1 The FFT graph $F^{(d)}$ can be pebbled in T moves with S pebbles where

$$T \leq \begin{cases} \dfrac{2n^2}{2^j} + (j-1)n & \text{for } S \geq d+2^j-j,\ 1 \leq j \leq d-1 \\ n(1 + \log_2 n) & \text{for } S \geq 2^d + 1 \end{cases}$$

and $n = 2^d$.

The upper bound grows more rapidly than n log n when S, the number of pebbles used, is o(n/(log n)) and grows as n^2 when $S = S_{min} + O(1)$ & $S_{min} = 1 + \log_2 n$. The upper bound is $T \leq n^2$ when $S = 1 + \log_2 n$, which cannot be improved upon by more than a factor of two, as shown in the next section. The upper bound cannot be improved upon for $S \geq 2^d + 1$ because it is then equal to the number of nodes in $F^{(d)}$.

The upper bound is stated only for certain values of S but it can be improved when $S \neq d + 2^j - j$ for any j, by extending the pebbling strategy given above. And of course the bound for $S = d + 2^j - j$ applies to $S \geq d + 2^j - j$ since the remaining pebbles need not be used.

We now turn to the derivation of lower bounds.

LOWER BOUNDS ON SPACE-TIME EXCHANGES

The key to the derivation of lower bounds on T given $S = d + 2^j - j$ is the observation that in the pebbling strategy given above, exactly one move is made in each node of $A_\ell^{(j)}$, $0 \leq \ell \leq 2^{d-j} - 1$, that is, nodes at level d-j or above, but that most nodes of $B_m^{(d-j)}$, $0 \leq m \leq 2^j-1$, or nodes at levels less than d - j have many moves made on them, that is, they are recomputed many times. An argument based on this observation is presented but is preceded by an argument for the special case of S = d + 1. This will allow us to introduce a number of ideas that are useful in the later presentation.

Paterson and Hewitt [3] have shown that at least k pebbles are necessary to pebble a directed, complete binary tree with 2^k leaves and which has the root as its only output. Their argument proceeds as follows: Consider the last point in time at which a path from inputs to the root of the tree is pebble-free. Branching from this path are k distinct paths to inputs each of which must be closed by the placement of at least one pebble. Thus, at least k + 1 pebbles must be used to close each of the branching paths and the path itself and thus to pebble the output. It should also be clear that k + 1 pebbles are sufficient. The time at which the last remaining open path in a complete binary tree is closed by a pebble will be called a *critical time* and the event will be called a *critical event*.

From Lemma 2 we know that the paths to the inputs from any given output of $F^{(d)}$ form a complete binary tree with 2^d leaves. Therefore, at least $S = d + 1$ pebbles are necessary to pebble $F^{(d)}$. Consider the first output for which a critical event occurs when pebbling $F^{(d)}$ with d+1 pebbles. The arrangement of pebbles is fixed at this time, as indicated in Figure 2. Note that exactly two outputs have their last open paths closed at the same critical time. Furthermore, exactly one of the d + 1 pebbles lies in ${}_0F^{(d+1)}$ or ${}_1F^{(d-1)}$ and this one pebble can only be used to pebble the two indicated output nodes. Also there are open paths to the remaining outputs. Thus, at a next critical time the relevant portion of the subgraph ${}_iF^{(d-1)}$ that contained the single pebble will have to be pebbled, so that either one pebble is brought to level d - 1 in $2^d - 1$ moves or d pebbles are brought to levels 0, 1, 2,..., d - 2 with two at level 0 and one at each remaining level. The number of moves made here is $1 + \sum_{j=0}^{d-2} (2^{j+1} - 1) = 2^d - d$.

In summary, when $S = d + 1$ pebbles are used to pebble $F^{(d)}$, exactly two outputs will have critical events at the same time so a series of 2^{d-1} distinct critical times will occur with at least $2^d - 1 + 2^d - d$ moves made to reach the first critical time and at least $2^d - d$ moves made to reach each of the remaining critical times. Consequently, at least $2^{d-1}(2^d-d) + 2^d - 1$ moves are made in all.

Theorem 2 At least $T \geqslant 2^{d-1}(2^d-d) + 2^d - 1$ moves are necessary to pebble $F^{(d)}$ with the minimum number of pebbles, $S = d + 1$.

Comparing Theorems 1 and 2 it is apparent that this lower bound differs from the upper bound by a factor of about two when d is large. The generalization of this argument is given in detail below but in summary form here. If S pebbles are available, we choose the integer j which satisfies

$$2^{j-1} + d - (j-1) < S \leqslant 2^j + d - j$$

We represent $F^{(d)}$ as given in Lemma 1, that is, in terms of the FFT graphs $A_\ell^{(j)}$, $0 \leqslant \ell \leqslant 2^{d-j}-1$, and $B_m^{(d-j)}$, $0 \leqslant m \leqslant 2^j-1$. We then examine a "snapshot" of $F^{(d)}$ at a critical time for some output and show that if there are a pebbles on $\{A_\ell^{(j)}\}$, at most 2^{j+1} outputs have paths blocked by pebbles to a/2 or more inputs of $\{A_\ell^{(j)}\}$. This implies that at least about 2^d moves have to be made on nodes in $\{B_m^{(d-j)}\}$ to go from the given critical time to a critical time for one of the $(2^d - 2^{j+1})$ outputs of $F^{(d)}$ which have less than a/2 paths to inputs of $\{A_\ell^{(j)}\}$ blocked by pebbles. Since a critical event must occur for each of the 2^d outputs, at least $2^d/2^{(j+1)}$ distinct critical events must occur with the property that to go from one to the other requires about 2^d moves on nodes in $\{B_m^{(\ell)}\}$. This is the essence of the lower bound argument.

Let $F^{(d)}$ be represented as in Lemma 1 with j chosen to satisfy

$$2^{j-1} + d - (j-1) < S \leqslant 2^j + d - j$$

where S is the maximum number of pebbles used to pebble $F^{(d)}$. At a critical time for some output a previously open path is closed. Suppose this path passes through $A_{\ell_0}^{(j)}$ and $B_{m_0}^{(d-j)}$. Then at least d - j + 1 pebbles must reside on $B_{m_0}^{(d-j)}$ to close this portion of the path so at most

$$R = S - (d - j + 1)$$

pebbles lie elsewhere on $F^{(d)}$ and

$$2^{j-1} < R \leqslant 2^{j} - 1$$

Lemma 3 Let a pebbles reside on the subgraphs $A_{\ell}^{(j)}$, $0 \leqslant \ell \leqslant 2^{d-j}$, of $F^{(d)}$. Let c_i be the number of inputs to these subgraphs whose paths to the i^{th} output of $F^{(d)}$ are blocked by pebbles. Then, the following inequality holds for $n = 2^d$:

$$\sum_{i=0}^{n-1} c_i \leqslant a2^{j}$$

Proof The graph $A_{\ell}^{(j)}$ is an FFT graph on 2^j inputs and Lemma 1 applies to it. From Lemma 2 a pebble placed at level k of $A_{\ell}^{(j)}$ blocks 2^k inputs for each of 2^{j-k} outputs. Thus, each pebble on $A_{\ell}^{(j)}$ contributes at most 2^j to the sum shown above from which the conclusion follows.

Lemma 4 Let N be the number of outputs of $F^{(d)}$ which have paths blocked by pebbles to $a/2$ or more inputs of $\{A_{\ell}^{(j)}\}$ when there are a pebbles on $\{A_{\ell}^{(j)}\}$. Then,

$$N \leqslant 2^{j+1}$$

Proof In Lemma 3, lower bound the sum by $N(a/2)$.

From the above two lemmas if a pebbles reside on $\{A_{\ell}^{(j)}\}$ at the critical time t for some output then at most $2^{j+1}-1$ additional outputs can have critical events before we reach the critical time for an output node v that has the property that at time t there were at most $\lfloor a/2 \rfloor$ paths blocked by pebbles from this node to inputs of $\{A_{\ell}^{(j)}\}$. If at time t b pebbles were on $\{B_m^{(d-j)} / m \neq m_o\}$ but below $\{A_{\ell}^{(j)}\}$, then it follows that $a + b \leqslant R \leqslant 2^j - 1$. But the nodes below $\{A_{\ell}^{(j)}\}$ form 2^{j+1} subgraphs $\{B_{m_1}^{(d-j-1)}\}$, as seen from Lemma 1. Thus, at most b + 1 of these subgraphs can have one or more pebbles on them. However, to pebble output node v we must close all paths through at least $2^j - \lfloor a/2 \rfloor$ inputs to $\{A_{\ell}^{(j)}\}$ or at least $2(2^j - \lfloor a/2 \rfloor)$ subgraphs $\{B_{m_1}^{(d-j-1)}, 0 \leqslant m_1 \leqslant 2^{j+1}-1\}$ of which at most b + 1 have pebbles at time t. Thus, at least $2^{j+1} - a - (b+1)$ of these subgraphs are pebble-free at time t and paths through them have to be closed to produce a critical event for node v. Since $a + b \leqslant 2^j - 1$, this applies that trees in at least 2^j distinct FFT graphs of d-j levels have to be closed in order to reach a critical event for v. (The FFT graph $F^{(d)}$ has d + 1 levels.)

In the pebbling of $F^{(d)}$ at least 2^d critical events must occur since each output is pebbled. Some of these events may occur at the same time. We identify a subset C of the critical events which occur at distinct times and count the number of moves necessary between these events. The first critical event is a member of this subset, as is the first subsequent critical event that is associated with an output with at most $\lfloor a/2 \rfloor$ paths to inputs of $\{A_{\ell}^{(j)}\}$ closed at that time. Other members of the subset C are defined in the same manner, that is, in terms of preceding critical events. It follows from Lemma 4 that the subset C contains at least $2^{d-(j+1)}$ critical events if $j \leqslant d - 1$ because at most $2^{j+1}-1$ critical events occur between events in this subset. At least one critical event occurs if $j \geqslant d$.

At this point we know that to pebble $F^{(d)}$ with S pebbles where

$$2^{j-1} + d - (j - 1) < S \leqslant 2^{j} + d - j$$

we must pebble all the nodes of $\{A_{\ell}^{(j)} | 0 \leqslant \ell \leqslant 2^{d-j} - 1\}$ at least once in at least $2^d(j+1)$ moves and must pebble $F^{(d)}$ below level d-j to reach a first critical event and must move enough pebbles between $2^{d-(j+1)}$ special critical events (if $j \leqslant d-1$) to block all paths through at least 2^j distinct pebble-free trees of d-j levels. To produce the first critical event in C we must block all paths in 2^{j+1} pebble-free trees of d-j levels. Paths in a tree can be blocked by pebbling all of the nodes in the tree or by pebbling a subset of the nodes, leaving a sufficient number of pebbles to block the open partial paths to the input.

Lemma 5 At least $n2^k - p$ moves are necessary to block all paths in each of n complete k-level binary trees using p pebbles.

Proof To pebble all nodes in a complete binary tree of k levels and $2^k - 1$ nodes, $2^k - 1$ moves are necessary. Suppose that a complete binary tree of k levels is partially pebbled and that q pebbles are left on the tree in positions so that all paths to the inputs are blocked. Then, the number of moves to block is minimized if these q pebbles are on the leaves of a binary subtree. It is easy to show that this subtree has q-1 internal nodes so at least $2^k - q$ moves are necessary to block all paths in a complete binary tree of k levels when q pebbles are left on the tree.

Consider pebbling n complete k-level binary trees using p pebbles. Let m be the number of trees that are completely pebbled and on which no pebbles are left and let r be the number of trees on which one or more pebbles is left, where $m + r = n$. Let $q_1, q_2, \ldots, q_r$ be the numbers of pebbles left on these trees. Then, at least

$$m(2^k - 1) + \sum_{i=1}^{r} (2^k - q_i) = n2^k - \sum_{i=1}^{r} q_i$$

moves are necessary. However,

$$\sum_{i=1}^{r} q_i \leq p$$

since all of the pebbles could be left on the trees, from which the desired conclusion follows.

We are now prepared to state the principal result of this section.

Theorem 3 The number of moves necessary to pebble the FFT graph $F^{(d)}$ on $n = 2^d$ inputs using $S < n$ pebbles satisfies

$$T \geq n(j + 2) + \frac{n(n - S)}{2 \cdot 2^j}$$

where $j \leq d$ is the integer determined by the following condition:

$$2^{j-1} + d - (j - 1) < S \leq 2^j + d - j$$

Proof It follows from the preceding discussion that $n(j+1)$ moves are necessary on $\{A_\ell^{(j)}\}$ and that a sufficient number of moves on $\{B_m^{(d-j)}\}$ are necessary below their outputs in order to block all paths in 2^{j+1} complete $(d-j)$-level binary trees for a first critical event and if $j \leq d - 1$, to block all paths in 2^j such trees for each of $2^{d-(j+1)} - 1$ remaining critical events. From Lemma 5 this means that at least

$$2^{j+1}(2^{d-j}) - S = 2 \cdot 2^d - S$$

moves are necessary for the first critical and at least

$$2^j(2^{d-j}) - S = 2^d - S$$

for each of the remaining events, for the desired conclusion.

Comparison of this result with the lower bound of Theorem 2 indicates that it is about a factor of two weaker than $S = d + 1$ and $j = 1$ and by comparison with the upper bounds of Theorem 1, the lower bound is no more than about a factor of four weaker than the upper bound when $S = 2^j + d - j$ for some $j \geq 1$.

CONCLUSIONS

The FFT algorithm on $n = 2^d$ inputs is an example of an important algorithm for which knowledge of the range of achievable space-time tradeoffs could prove quite useful.

We have derived lower bounds to the number of units of time necessary to pebble the graph $F^{(d)}$ which characterizes the uninterpreted FFT algorithm and have shown that time which is $O(n \log n)$ can only be achieved if S pebbles are used where $S = \Omega(n/\log n)$. If $S = o(n/\log n)$ then the time rises more quickly than $O(n \log n)$ with n and in the limit of $S = d + 1$ the time grows as $O(n^2)$. Thus, the savings offered by the FFT algorithm to compute the Fourier transform can be lost if insufficient temporary storage is not available.

The bounds of Theorems 1 and 3 can be recast as bounds on the expression

$$Q = (S - d)\ (T - nd)$$

From Theorem 1 it is easy to see that

$$Q \leqslant 4n^2$$

if we choose $1 \leqslant j \leqslant d - 1$ such that

$$2^j + d - j \leqslant S \leqslant 2^{1+1} + d - (j+1)$$

A lower bound to Q of the form

$$Q \geqslant n^2 \frac{11}{128} - \frac{nd}{8}$$

results if we choose $1 \leqslant j \leqslant d$ such that

$$2^{j-1} + d - (j-1) < S \leqslant 2^j + d - j$$

and lower bound S by 2^{j-2} and upper bound S by $2^j + d$. These hyperbolic bounds on space-time exchanges are similar to the hyperbolic bounds derived with the combinational complexity measure [10].

ACKNOWLEDGEMENT

The authors offer their thanks to Nick Pippenger for the suggestion that the bounds on space and time could be stated as hyperbolic inequalities over the entire range of space and time.

REFERENCES

1. Cooley, J. M. and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," Math. Computation, Vol. 19, pp. 297-301, 1965.
2. Cooley, J. M., P. A. Lewis, and P. D. Welch, "History of the Fast Fourier Transform," Proc. IEEE, Vol. 55, pp. 1675-1677, 1967.
3. Paterson, M. S. and C. E. Hewitt, "Comparative Schematology," Proj. MAC Conf. on Concurrent Systems and Parallel Computation, Woods Hole, MA, pp. 119-127, June 2-5, 1970.
4. Hopcroft, J. E., W. J. Paul, and L. G. Valiant, "On Time Versus Space," JACM, Vol. 24, No. 2, pp. 332-337, 1977.
5. Paul, W. J., R. E. Tarjan, and J. R. Celoni, "Space Bounds for a Game on Graphs," Eighth Ann. Symp. on Theory of Computing, Hershey, PA, pp. 149-160, May 3-5, 1976.
6. Pippenger, N., "A Time-Space Tradeoff," IBM preprint, May, 1977.
7. Chandra, A. K., "Efficient Compilation of Linear Recursive Programs," IBM Research Rept. RC4517, 10 pp. August 29, 1973.
8. Paul, W. J. and R. E. Tarjan, "Time-Space Tradeoffs in a Pebble Game," preprint, May, 1977.
9. Borodin, A. and I. Munro, The Computational Complexity of Algebraic and Numeric Problems, American Elsevier, New York, 1975.
10. Savage, J. E., The Complexity of Computing, Wiley-Interscience, New York, 1976, Chapter 7.

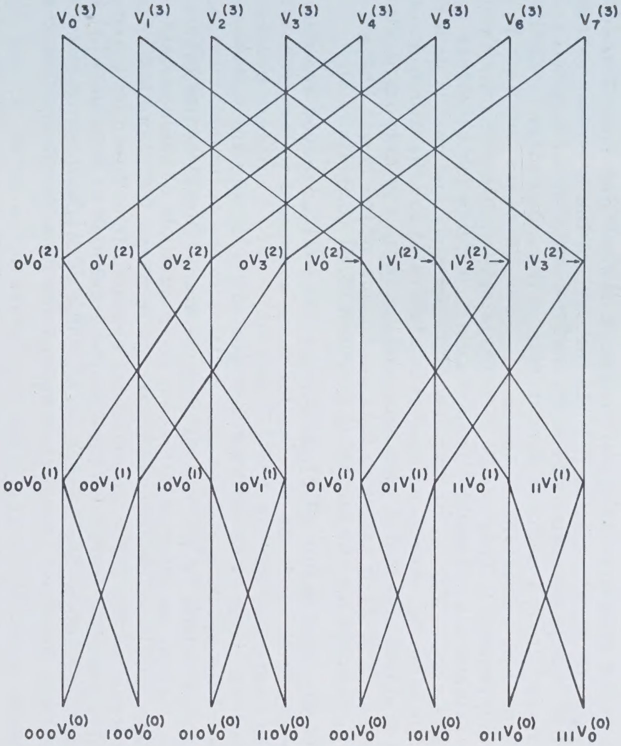

$V_0^{(3)}$
$V_1^{(3)}$
$V_2^{(3)}$
$V_3^{(3)}$
$V_4^{(3)}$
$V_5^{(3)}$
$V_6^{(3)}$
$V_7^{(3)}$
$_0V_0^{(2)}$
$_0V_1^{(2)}$
$_0V_2^{(2)}$
$_0V_3^{(2)}$
$_1V_0^{(2)}$
$_1V_1^{(2)}$
$_1V_2^{(2)}$
$_1V_3^{(2)}$
$_{00}V_0^{(1)}$
$_{00}V_1^{(1)}$
$_{10}V_0^{(1)}$
$_{10}V_1^{(1)}$
$_{01}V_0^{(1)}$
$_{01}V_1^{(1)}$
$_{11}V_0^{(1)}$
$_{11}V_1^{(1)}$
$_{000}V_0^{(0)}$
$_{100}V_0^{(0)}$
$_{010}V_0^{(0)}$
$_{110}V_0^{(0)}$
$_{001}V_0^{(0)}$
$_{101}V_0^{(0)}$
$_{011}V_0^{(0)}$
$_{111}V_0^{(0)}$

Fig. 1 FFT Graph with Eight Inputs

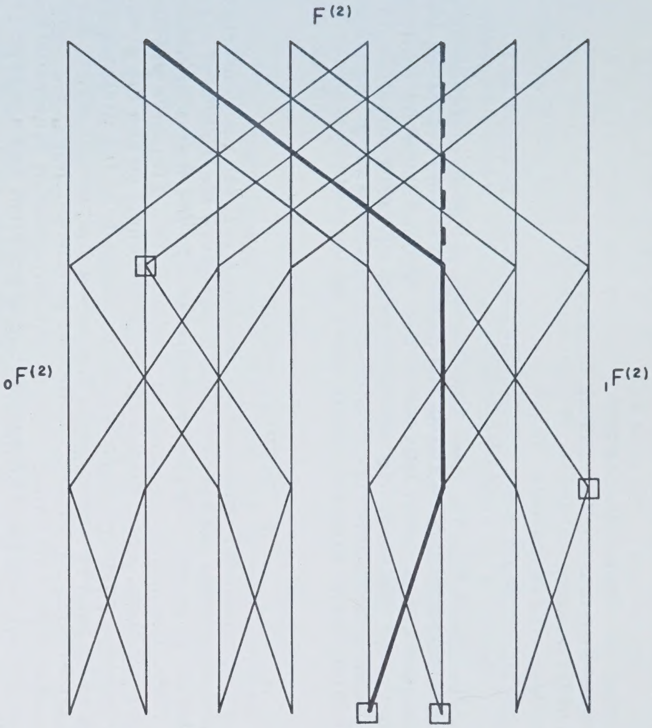


Fig. 2 Critical Event on an FFT Graph

THE PROBABILISTIC ANALYSIS OF MATCHING HEURISTICS*

CHRISTOS H. PAPADIMITRIOU
Center for Research on Computing Technology
Harvard University
Cambridge, Massachusetts 02138

ABSTRACT

It can be shown that the total length of the optimum matching of a set of n points uniformly distributed on the unit square--or any other Lebesgue area of measure 1 --is, when divided by $\sqrt{n}$, almost certainly equal to a constant $\mu \geq .25$. We analyze probabilistically several heuristics for the matching problem and obtain upper bounds on the value of μ. We show that $\mu \leq .40106$, and conjecture that $\mu \simeq .35$.

1. INTRODUCTION

Many combinatorial optimization problems call for the construction of the shortest possible network of some kind, given the matrix of distances among a set of n points. These problems include the traveling salesman problem (TSP), the minimum spanning tree problem, Steiner's problem, the matching problem, etc. (See [18] for definitions.) When the given points are actually realized as points on the plane--and the distance matrix is thus induced by the two-dimensional Euclidean metric--we obtain the Euclidean case of such a problem. Different problems, however, behave differently under this restriction. The minimum spanning tree problem becomes easier [12], but the TSP and Steiner's problem remain hard [5, 9]. There is no obvious advantage to the matching problem [11], and the complexity of another problem is reduced from probably exponential to O(n log n) [11].

For these Euclidean cases of "network design" problems a very strong probabilistic result is available [2]. In the case of the traveling salesman problem this result essentially asserts that the value of the optimal tour of n points drawn from a uniform distribution in the unit square is almost certainly equal to a constant β times $\sqrt{n}$. (See Theorem 1 of Section 2 for a precise statement.) This result is surprisingly stable when the assumptions are relaxed significantly. In particular, it holds for arbitrary measurable regions with arbitrary probability distributions, and can be generalized to many dimensions. Furthermore, it is also valid for the minimum spanning tree problem, Steiner's problem, and, in fact, any network design-type of problem as long as it satisfies four conditions stated in our Section 2. In particular, it holds for the matching problem [3, 8] (Corollary 2).

Recently, the result of [2] received some well-deserved attention from computer scientists and was used as the main argument in the probabilistic analysis of heuristics for the TSP [7] and other hard Euclidean problems [10]. However, heuristics such as the ones in [7, 10] are susceptible to evaluation of the solution yielded only relative to the optimum solution, and not in absolute terms (e.g., calculating the actual length of the derived solution). Hence, no information about the magnitude of the constant β in the theorem of [2] can be deduced by analyzing them. We also notice here that in contrast to the TSP, the matching problem (Euclidean or otherwise) can be solved exactly in $O(n^3)$ time by the algorithm of Edmonds [3] as implemented by Gabow and Lawler [4, 8]. The conceptual complexity of this algorithm, however, renders it beyond any detailed probabilistic analysis.

---

*The research reported here was supported by NSF Grant MCS 77-01193.

Determining the constant β is an interesting problem in its own right. In [2] the bounds $.61 \le \beta \le .92$ are proved. Explicit calculation of β --or even an improvement on the above bounds--appears to require a significant departure from the existing methodology. Monte Carlo experiments in [2] estimate β to be .75; based on more elaborate experiments, D. Stein [13] estimates it to .765.

The value of the corresponding constant for the matching problem, μ, is unknown as well. Since the optimum matching is at most a half of the shortest tour, we notice immediately that $\mu \le .5\beta \le .46$. We also show that $\mu \ge .25$. The last part of this paper concentrates on improving the .46 upper bound using techniques that are applicable to matching and not to the TSP. For this purpose we analyze probabilistically certain matching heuristics, most notably a three-phase extension of the "monotonic" heuristic. The constant corresponding to the three-phase heuristic is shown to be less than .40106.

2. THE SHORTEST PAIRING OF MANY POINTS

We will state the result of [2] for two dimensions, the unit square and uniform distribution. The possible extensions were sketched in the Introduction.

Let $P = (p_1, p_2, \dots)$ be a random variable with range infinite sequences of points independently and uniformly distributed in the unit square. Let $P^n = \{p_1, \dots, p_n\}$, and let $T(P^n)$ denote the shortest tour of all the points in P^n.

THEOREM 1 [2]. With probability 1, the limit $\lim_{n\to\infty} T(P^n)\cdot n^{-1/2}$ exists and is equal to a constant β. ■

The proof of Theorem 1 essentially consists of two parts. First, the TSP is shown to satisfy four simple combinatorial conditions. Let Q be a finite set of points in an area A, and let $A_1, \dots, A_m$ be a partition of A. $Q_1, \dots, Q_m$ is the corresponding partition of Q. T satisfies the following conditions:

1. $$T(Q) \le \sum_{j=1}^{m} (T(Q_j) + \delta_{j,j+1}) ,$$

where δ_{ij} is the diameter of $A_i \cup A_j$ and $\delta_{m,m+1} = \delta_{m1}$.

2. $$T(Q) \ge \sum_{j=1}^{m} (T(Q_j) - 2\cdot|\partial A_j|) ,$$

where $|\partial A_j|$ is the length of the boundary of A_j.

3. There exists a constant α such that $T(P^n) \le \alpha\sqrt{n}$.

4. There exists a constant γ such that $\mathscr{E}(T(P^n)) \ge \gamma\sqrt{n}$.

The second part of the proof involves a deep analytic argument which uses only properties (1-4) of the TSP. An elementary exposition of some of the argument appears in [11].

Let P be as in Theorem 1. A matching of P^n is a set of line segments (p_{i_k}, p_{j_k}), $k = 1, \dots, \lfloor n/2 \rfloor$ such that all i_k, j_m are distinct. Let $M(P^n)$ be the shortest such matching.

COROLLARY 2. With probability 1, the limit $\lim_{n\to\infty} M(P^n)$ exists and is equal to a constant μ.

Proof. We will prove properties (1-4) with T replaced by M. The corollary will then follow from the proof of Theorem 1 given in [2].

1. Let $k_1, \ldots, k_{q'}$ be the indices j for which $|Q_j|$ is odd. If q' is itself odd, consider $q = q'-1$; otherwise, $q = q'$. The optimal matchings of Q_j together with the segments $(p_{k_1}, p_{k_2}), \ldots, (p_{k_{q-1}}, p_{k_q})$ is a matching of Q, hence

$$M(Q) \leq \sum_{j=1}^{m} M(Q_j) + \sum_{j=1}^{q/2} \delta_{k_{2j-1}, k_{2j}} \leq \sum_{j=1}^{m} (M(Q_j) + \delta_{j,j+1}) .$$

2-3. Precisely as in Lemmas 1 and 4 of [2]. Instead of 2, we can show the stronger

$$M(Q) \geq \sum_{j=1}^{m} \left(M(Q_j) - \frac{1}{2} |\partial A_j| \right) .$$

4. Consider a point $p \in P^n$, and let p' be a point in P^n different from but closest to p. Let $d(p,p')$ be the Euclidean distance between p and p'.

To calculate $s(p) = \mathscr{E}(d(p,p'))$, we notice that $\text{prob}(d(p,p') > r) = (1 - A_p(r))^{n-1}$, where $A_p(r)$ is the area of the intersection of the unit square and the disc of center p and radius r. Thus,

$$\begin{aligned} s(p) &= \int_0^\infty (1-A_p(r))^{n-1}\, dr \geq \int_0^1 (1 - \min(1, \pi r^2))^{n-1}\, dr \\ &= \frac{1}{\sqrt{\pi}} \int_0^1 (1-x^2)^{n-1}\, dx \\ &= \frac{\Gamma(n)}{2\Gamma\left(n+\frac{1}{2}\right)} \geq \frac{.25}{\sqrt{n}} . \end{aligned}$$

Since a matching of P^n at best will pair each $p \in P^n$ with its closest neighbor, we have that

$$\mathscr{E}(M(P^n)) \geq \frac{1}{2} \mathscr{E}\left(\sum_{p \in P^n} s(p) \right) = \frac{1}{2} \sum_{p \in P^n} \mathscr{E}(s(p)) \geq .25\sqrt{n} . \blacksquare$$

Notice that from the last inequality it follows that $\mu \geq .25$. In Section 5 we show that $\mu \leq .40106$.

3. THE MONOTONIC HEURISTIC

The probabilistic analyses of this and the next section will fall into the following framework. Let $\mathscr{A}$ be an algorithm which, given a set of points on the plane, produces a feasible matching. Consider the algorithm applied to sets of points on a $\xi \times \xi$ square drawn by a unit density Poisson process. Let $2\mu_{\mathscr{A}}$ be the limit of the average length of a matched segment, as $\xi \to \infty$. Then it follows that $\mu_{\mathscr{A}} \geq \mu$. (This is a major part of the argument in [2].)

Our first algorithm is the monotonic heuristic (Fig. 1). In this algorithm we divide the square into vertical strips of width w. In each strip we scan the points bottom-up, matching each free point that we meet with the one with the next higher y-coordinate. The point that is possibly left unmatched in each strip does not influence our calculations, since its contribution is asymptotically negligible (recall that $\xi \to \infty$).

This heuristic is essentially the same as the one analyzed in [2], Lemma 10; following their analysis, we may determine $\mu_M(w)$, the $\mu_{\mathscr{A}}$ for the monotonic algorithm with width w. The least value of $\mu(w)$ is .46, attained for $w^2 = 3$. Notice that the .46 bound is the same as that derived in the Introduction using an independent argument.

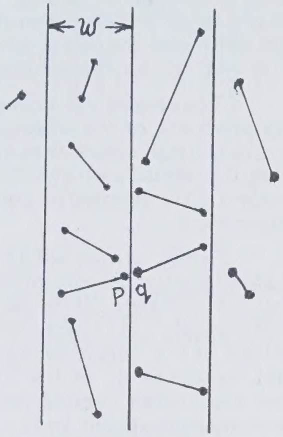


Figure 1

A version of this monotonic algorithm is the best known TSP heuristic, whose average performance is analytically tractable (still, its analysis in [2] resorts to numerical quadrature and special function calculations). We will see in the next section that the matching problem, by being more relaxed combinatorially, will allow the probabilistic analysis of more effective heuristics, and an improvement on the .46 upper bound for μ.

4. THE THREE-PHASE ALGORITHM

The algorithm whose analysis will yield our upper bound for μ is an elaborate implementation of the monotonic heuristic. For the purposes of clarity we first present the basic ideas in the algorithm and then describe the modifications necessary in order to improve its effectiveness and analytic tractability.

As in the monotonic heuristic, we divide the square into strips of width w. In the monotonic heuristic all points in a strip--except for possibly one--are matched to points in the same strip. Thus, points that are very close to each other but in different strips (the points p, q in Fig. 1) cannot be matched together. To avoid such anomalies, our new algorithm processes the points in three phases. In the first phase, we process the points in the column of width 2d around the dividing line of two adjacent strips (see Fig. 2). Each free point in these columns is paired with the one having immediately higher y-coordinate (these are the points marked 1 in Fig. 2). However, if this next point is more than a vertical distance L away (as is the case of the points marked 2 in Fig. 2) we leave this point unmatched and proceed to the next point. w, d and L are parameters.

In the second phase, we match the points in the columns that were left unmatched in the first phase. We match them with the point in the main strip (in fact, in the corresponding left and right halves of the two neighboring main strips) that has the next greater y-coordinate. (These are the points marked 2' in Fig. 2.) Finally, in the third phase we match the remaining points in the main strips (marked 3 in Fig. 2) in the monotonic fashion.

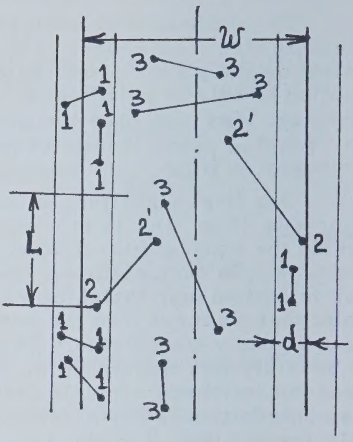


Figure 2

There are certain difficulties in the analysis of the algorithm as stated. We briefly discuss these, together with the modifications that result in an analytically tractable and effective algorithm.

First, we would like the y distance between the points of type 2 and their mates 2' to be exponentially distributed. It is not, however. The reason is that two successive type-2 points in the same column may interact in a non-trivial way. It may be that the mate 2' of the lowest is above the higher type-2 point. This means that the higher type-2 point will not be matched with the next--in terms of y coordinates--point in the main strip, but to the second next. The distribution is not Poisson.

We can avoid this difficulty by an easy modification of the algorithm. Whenever the mate 2' of a type-2 point is further--with respect to y coordinates--than the next point of the column, then we match the type-2 point at hand not to the 2' point, but to the next point in the column. It is easy to see that this does not increase the expected value of the total matching. This has the effect of "decoupling" successive type-2 points and rendering the distribution of the y-lengths of matched segments of phase 2 exponential. Note that in order to implement this modification, phases 1 and 2 must be interleaved.

In phase 3 the algorithm matches monotonically the points that were left unmatched in phases 1 and 2 (type 3 points). These points are, roughly speaking, the sum of two Poisson processes: the original one of unit density, and another process corresponding to the type-2 points. The latter process is almost Poisson--with a "dead" period of length L following each event. The effect of this process is that each even in it cancels the event of the original process that comes after it. An alternative formulation would be via a simple marked process [14]. Unfortunately, the resulting process of the surviving points is not Poisson. Quite to the contrary, the interarrival interval distribution function Laplace-transforms to a complicated irrational function, and it is not at all clear even how it could be approximated by a Poisson process.

To overcome this difficulty, we modify phase 3 of our algorithm. For each type-3 point--i.e., a point left unmatched by phases 1 and 2--we scan the entire strip by increasing y's. If a type-3 point is met before a type-2 point (Fig. 3a), then the two points are matched (dotted segment). However, if a type-2 point is met first (Fig. 3b), the matching is changed as shown in Fig. 3c. We call such points type-4, to distinguish from the type 3 points of case (a). The mate of the type-2 point is now considered as an ordinary type-3 point (Fig. 3c). The result of this modification is that the

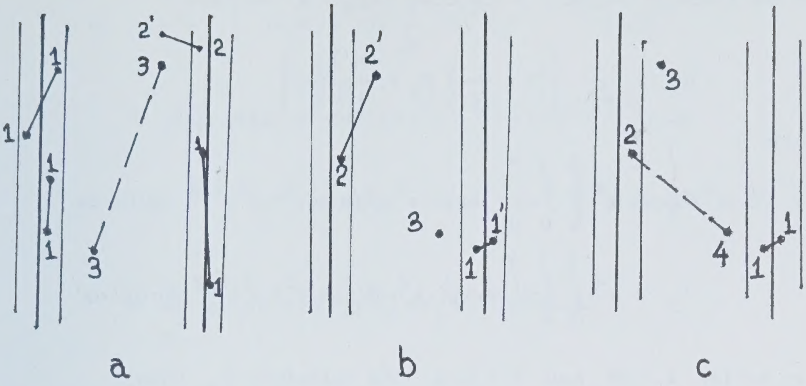


Figure 3

distribution of the y distances between two matched type-3 points is now exponential. Notice that this modification, unlike the previous one, has the effect of worsening the average overall length of the matching produced; its main purpose is to ease the analysis that follows.

5. ANALYSIS OF THE THREE-PHASE ALGORITHM

Let $2 \cdot \mu(w, d, L)$ be the average value of a matched segment for the three-phase algorithm of the previous section. In this section we will bound $\mu(w, d, L)$ from above. The derivation is complicated and lengthy, and due to space constraints, we will present here a sketch of the basic ideas involved. The methodology is, generally, analytical using approximations and special functions, and the trade-off is between accuracy and analytical tractability.

In order to estimate $\mu(w, d, L)$, we shall first calculate the densities r_1, r_2, r_3, r_4 of the points of the different types, and the average matched half-segments between points of the different types μ_1, μ_2, μ_3.

5.1 Phase 1

The process of scanning a column in phase 1 and marking its points 1 or 2 can be envisioned as a Markov process with the state diagram shown in Fig. 4. State 2 corresponds to a type-2 point; state 1 corresponds to a type-1 point that is the lower end of the matched segment, and state 1' corresponds to the rest of type-1 points. q' is the probability that, given a point in the column, the next point in this column will be the next point in the corresponding half of the strip. q' can be easily calculated as $q' = q \cdot (1 - 4\exp(-wL/2)\, d/w)$, $q = \exp(-2dL)$. The analysis of the state diagram of Fig. 4 gives:

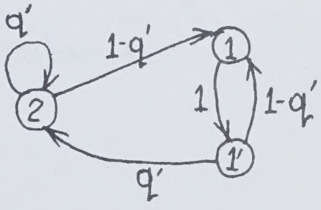


Figure 4

$$r_1 = \frac{4d(1-q')}{w(2-q')}, \qquad r_2 = \frac{4dq'}{w(2-q')}.$$

For the average length of a matched segment, we have

$$\mu_1 = \frac{1}{2a}\left[\left(1 - \frac{q'}{q}\right) J_1 + \frac{q'J_2}{q(1-q)}\right] \quad ,$$

where

$$J_1 = a^4 \int_0^1 \int_0^1 \int_0^\infty \exp(-a^2 y)[(x-x')^2 + y^2]^{1/2} \, dy \, dx' \, dx$$

$$J_2 = a^4 \int_0^1 \int_0^1 \int_0^k \exp(-a^2 y)[(x-x')^2 + y^2]^{1/2} \, dy \, dx' \, dx \quad .$$

Here we let $a = 2d$ and $k = L/a$. We calculate J_2 first.

$$J_2 = 2a^2 \int_0^1 (1-u) \int_0^k \exp(-a^2 y)[u^2 + y^2]^{1/2} \, du \, dy$$

$$= 2a^2 \left[\frac{1}{6} - q\left(\frac{\sqrt{1+k^2}}{2} + \frac{k^2}{2} \ln\left(1 + \sqrt{1+k^2}\right) - \frac{(1+k^2)^{3/2}}{3} - \frac{k^2 \ln k}{2} + \frac{k^3}{3}\right)\right] + I_1 + I_2$$

$$I_1 = 2a^2 \int_0^1 u \int_0^{k/u} \exp(-a^2 uz) z (1+z^2)^{1/2} \, dz \, du$$

$$= 2 \int_0^1 u \int_0^{k/u} \frac{\partial}{\partial u} [\exp(-a^2 uz)(1+z^2)^{-1/2}] \, dz \, du$$

$$= 2kq \int_0^1 \frac{du}{(u^2+k^2)^{1/2}} - 2 \int_0^k \exp(-a^2 z)(1+z^2)^{1/2} \, dz - 2 \int_0^1 \int_0^{k/u} \frac{\exp(-a^2 uz)}{(1+z^2)^{1/2}} \, dz \, du$$

$$\leq -2kq\left[\ln\left(1+\sqrt{1+k^2}\right) - \ln k\right] - I_3 + \frac{\pi}{a^2} \int_0^a (H_0(t) - Y_0(t)) \, dt - 2E_1(a^2 k)$$

$$I_2 = -2a^2 \int_0^k \exp(-a^2 y) y \int_0^1 u(u^2+y^2)^{-1/2} \, du\, dy$$

$$= -2 \int_0^k \exp(-a^2 y)(2(1+y^2)^{1/2} - (1+y^2)^{-1/2}) \, dy$$

$$- 2a^2 \int_0^k y \exp(-a^2 y) \, dy$$

$$\leq \frac{2\pi}{a^2} (H_1(a^2)+Y_1(a^2)) + I_3 + \frac{4 - 2qk^2a^4}{a^4}$$

Here H_0, H_1 are Struve functions, Y_0, Y_1 are Bessel-2 functions and E_1 is the exponential integral. Then

$$J_2 \leq 2kq\left(\ell n\left(1+\sqrt{1+k^2}\right) - \ell n\, k\right) + \frac{\pi\phi(a^2)}{a^2} - 2E_1(a^2k)$$

$$+ \frac{4 - 2qk^2a^4}{a^4} + 2a^2\left[\frac{1}{6} - q\left[\frac{\sqrt{1+k^2}}{2} + \frac{k^2}{2}\left(\ell n\left(1+\sqrt{1+k^2}\right) - \ell n\, k\right)\right.\right.$$

$$\left.\left. - \frac{1}{3}((1+k^2)^{3/2} - k^3)\right]\right]$$

By $\phi(x)$ we denote the function $\int_0^x (H_0(t)-Y_0(t))\,dt - 2(H_1(x)-Y_1(x))$; values of ϕ calculated from [1] are tabulated below

x	2	2.5	3	3.5	4
$\phi(x)$	-.206689	-.006253	.144263	.264366	.364285

J_1 can now be calculated from J_2 with $k \to \infty$, $q = \exp(-a^2k)$.

$$J_1 = \frac{\pi\phi(a^2)}{a^2} + \frac{4}{a^4} + \frac{a^2}{3} \quad .$$

5.2 Phase 2

Let $\mu_{21}(\mu_{22})$ be the expected length of a phase-2 segment, given that the two mates are in the same strip (Fig. 5a) (resp. different strips, Fig. 6b). They can be estimated by

$$\mu_{21}^2 \leq \frac{1}{4cd}\int_0^c\int_0^d (x+x')^2\,dx'\,dx + \frac{c}{2}\int_0^\infty \exp(-2cy)y^2\,dy$$

$$= \frac{(c+d)^4 - c^4 - d^4}{48dc} + \frac{1}{8c^2}$$

$$\mu_{22}^2 \leq \frac{1}{4cd}\int_0^c\int_0^d (x+x'+d)\,dx'\,dx + \frac{c}{2}\int_0^\infty \exp(-2y)y^2\,dy$$

$$= \frac{(2d+c)^4 - (2d)^4 - (d+c)^4 + d^4}{48dc} + \frac{1}{8c^2} ,$$

where $c = (w/2) - d$. Finally, $\mu_2 = \frac{1}{2}(\mu_{21} + \mu_{22})$.

5.3 Phase 3

An unmatched point during phase-3 a type-3 or type-4 point depending on whether the next (in increasing y's) unmatched point comes before a type-2 point. Since both events have exponentially distributed times of first occurrence, it follows that

$$r_3 = 1 - r_1 - r_2 , \quad r_4 = \frac{2c}{w} \cdot \frac{q''}{2c+q''} ,$$

where

$$q'' = \frac{dq'}{2-q'} .$$

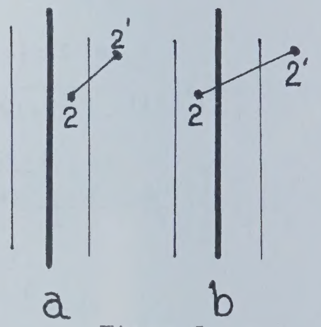


Figure 5

$$\mu_3 = \frac{(2c)^3}{2}\int_0^1\int_0^1\int_0^\infty \exp(-4c^2y)[(x-x')^2+y^2]^{1/2}\,dx'\,dx\,dy \quad ;$$

this is identical to the integral J_1 of 5.1. Hence

$$\mu_3 = \frac{\pi\cdot\phi((2c)^2)}{2(2c)^3} + \frac{2}{(2c)^5} + \frac{(2c)^2}{6} .$$

Finally, each type-4 point increases the estimate of μ_2 of 5.2 from $\mu_2 = \frac{1}{2}(\mu_{21} + \mu_{22})$ to $\mu_2' = \frac{1}{2}(\mu_{22} + \mu_{23})$, where

$$\mu_{23}^2 \leq \frac{1}{4cd}\int_0^c\int_0^d (x+x'+d+c)^2\,dx'\,dx + \frac{c}{2}\int_0^\infty \exp(-2cy)y^2\,dy$$

$$= \frac{(2d+2c)^4 - (2d+c)^4 - (2c+d)^4 + (c+d)^4}{48cd} + \frac{1}{8c^2} .$$

5.4 Computation

For values of the Struve and Bessel-2 functions we used [1]. For the exponential integral we used

$$E_1(x) \simeq \frac{1}{x\exp(x)}\ \frac{x^2+2.3347x+.2506}{x^2+3.3306x+1.6815} .$$

For $w = 2.6$, $d = .55$ and $L = 2.31$, the value of $\mu(w,d,L)$ is found .4010552.

6. REMARKS

Monte Carlo experiments on problems of size up to 200 points uniformly distributed on the unit square suggest that $\mu \simeq .35$. The results are shown below.

n	20	40	40	60	80	100	118	194
$M(P^n)/\sqrt{n}$	.38955	.319232	.3407	.3808	.3878	.3702	.37882	.36177

Table 2

A greedy algorithm can be devised to solve the matching problem as follows: process the points in order of increasing y's. Match each free point with the closest free point that has an x-distance from the one considered w or less. (w is a parameter.) The analysis of this algorithm is much harder. There is a non-trivial interaction between subsequent matchings, and it appears that the only way to estimate the performace of the greedy algorithm is by assuming the worst possible such interaction. The bound thus achieved is .675. By a more elaborate analysis--using conditional probabilities on the existence of previously matched points in certain regions--a bound of .49 is achieved. Despite the fact that the bounds we were able to prove about the greedy algorithm are inferior to the ones proved for other algorithms, this heuristic appears to perform much better when applied to random instances.

Improved upper bounds for matching is a problem that is certainly left open here. For the TSP, improving on the monotonic heuristic and the .92 upper bound appears to be extremely hard. Incidently, one cannot fail to notice that the gap between the conjectured constant and the best known upper bound appears to be increasing with the algorithmic complexity of the problem. For the spanning tree problem, this gap is just from .68 to .707 [6]; for matching--certainly a problem harder than the minimum spanning tree problem--we pointed out a .35 to .401 gap; finally, for the notoriously difficult TSP, the obscure area spans .765 to .92. Naturally, this is certainly due to the algorithmic techniques for proving upper bounds that are the only ones available to us now.

All known lower bounds for the constant of the TSP and other Euclidean network design problems are all derived by arguments as primitive as the one in the proof of Corollary 2. Improved lower bounds may require totally different techniques.

References

1. M. Abramowitz, I.A. Stegun, Handbook for Mathematical Functions, Dover, 1965.

2. J. Beardwood, J.H. Halton, J.N. Hammersley, "The Shortest Path Through Many Points", Proc. Camb. Phil. Soc. 55, 4 (1959), pp. 299-326.

3. J. Edmonds, "Paths, Trees, and Flowers", Canad. Math. J. 17 (1965), pp. 449-465.

4. H. Gabow, E.L. Lawler, "An Efficient Implementation of Edmonds' Algorithm for Maximum Matching on Graphs", TR No. 075, Dept. of Computer Science, University of Colorado, 1975.

5. M.R. Garey, R.L. Graham, D.S. Johnson, "Some NP-Complete Geometric Problems", Proc. 8th ACM Symp. on the Theory of Computing, Hershey, Pa., May 1976.

6. E.N. Gilbert, "Random Minimal Trees", J. SIAM 13, 2 (1965), pp. 376-387.

7. R.M. Karp, "The Probabilistic Analysis of Some Combinatorial Search Algorithms", in J.F. Traub (ed.), Algorithms and Complexity, Academic Press, 1976.

8. E.L. Lawler, Combinational Optimization: Networks and Matroids, Holt, Rinehart and Winston, 1976.

9. C.H. Papadimitriou, "The Euclidean TSP is NP-Complete", J. Th. Comp. Sci., to appear, 1977.

10. ________, "The Complexity of the Capacitated Tree Problem", Networks, to appear, 1978.

11. C.H. Papadimitriou and K. Steiglitz, Combinatorial Optimization Algorithms, manuscript, 1977.

12. M.I. Shamos, "Geometric Complexity", Proc. 7th ACM Symp. on the Theory of Computing, pp. 224-233, 1975.

13. D.M. Stein, Scheduling Dial-a-Ride Transportation Systems: An Asymptotic Approach, Ph.D. thesis, Harvard University, 1977.

14. J. Snyder, Random Point Processes, Wiley, 1975.

APPROXIMATE SOLUTIONS FOR THE TWO-PARTITION PROBLEM

CHUL E. KIM
Department of Computer Science, University of Maryland
College Park, Maryland 20742

Abstract SIMPLE-SPLIT and LPT algorithms for the two-partition problem are analyzed in terms of their probabilistic performance. The expected relative errors are shown to be proportional to $n^{-3/2}$ and n^{-2}, where n is the problem size. The results of computational experiments are given. Also an ε-approximation algorithm of time and space complexity of $O(n) + O(\varepsilon^{-2})$ is presented.

INTRODUCTION

Consider the two-partition problem which may be described as follows:

A set $S = \{s_1, s_2, \ldots, s_n\}$ of real numbers in (0,1) is given.

Partition S into P and Q such that $C = \max\{\sum_{s\varepsilon P} s, \sum_{s\varepsilon Q} s\}$ is minimum.

This problem is NP-hard [1,7] and hence it is unlikely that there exists a polynomial time algorithm to solve it. Thus we consider algorithms that solve the problem efficiently but only approximately.

Let A be an approximation algorithm, and let C^* and C_A be the optimal solution and the solution by A, respectively. The performance of A may be measured in terms of either the worst-case bound of C_A/C^* [2,5,6,9,10] or $E(C_A/C^*)-1$, the expected relative error [4,8].

First, two simple heuristic algorithms, SIMPLE-SPLIT and LPT, are analyzed. It can be shown that the worst-case bound for SIMPLE-SPLIT is 3/2. Graham showed that for the LPT algorithm, the worst-case bound is 7/6 [3]. In this paper the expected relative errors are shown to be proportional to $n^{-3/2}$ and n^{-2}, respectively, under certain assumptions on the probability distribution of the sample space. Results of computational experiments are given, and seem to confirm the analysis. Next, an ε-approximation algorithm is presented. It is shown that for any given ε, $0<\varepsilon<1$, the algorithm has the time and space complexity $O(n)+O(\varepsilon^{-2})$ and yields a solution C_ε satisfying $C_\varepsilon/C^* < 1+\varepsilon$.

SIMPLE-SPLIT ALGORITHM

A heuristic algorithm SIMPLE-SPLIT is described below:

1. Sort S in nonincreasing order such that $s_1 \geq s_2 \geq \cdots \geq s_{4m+k}$, where $n = 4m+k$ and $0 \leq k \leq 3$.
2. Partition S into P_S and Q_S as

$$P_S = \{s_1, s_3, \ldots, s_{2m-1}, s_{2m+2}, s_{2m+4}, \ldots, s_{4m}, s_{4m+1}, s_{4m+3}\}$$
$$Q_S = \{s_2, s_4, \ldots, s_{2m}, s_{2m+1}, s_{2m+3}, \ldots, s_{4m-1}, s_{4m+2}\},$$

This research was supported in part by NSF under Grant MCS76-82082.

where $s_{4m+\ell}=0$ if $\ell>k$.

We will denote $\sum_{s\varepsilon R} s$ by c(R), where $R \subseteq S$.

For the probabilistic analysis, we make the following assumptions:

Each element in S is drawn independently from a uniform distribution over (0,1). (1)

$\{s_1,s_2,\ldots,s_{2m-1}\}$ and $\{s_{2m+1},s_{2m+2},\ldots,s_{4m+k}\}$ are uniformly distributed over $(s_{2m},1)$ and $(0,s_{2m})$, respectively. (2)

Assumption (2) is restrictive. However, for the analysis of this algorithm the samples with assumptions (1) and (2) may reasonably approximate the samples with assumption (1) only.

By assumption (1), the event that $x<s_{2m}<x+\delta x$ for some $x\varepsilon(0,1)$ has the probability

$$(4m+k)!/((2m-1)!(2m+k)!)\ x^{2m+k}\ (1-(x+\delta x))^{2m-1}\ \delta x + O(\delta x^2),$$

where the lower order term $O(\delta x^2)$ represents the events that some other elements are in $(x,x+\delta x)$.

By assumption (2),

$$\begin{aligned} T = (1/2)\textstyle\sum_{s\varepsilon S} s &= (1/2)(\textstyle\sum_{i=1}^{2m} s_i + \sum_{i=2m+1}^{4m+k} s_i) \\ &= (1/2)(2m\cdot s_{2m}+((2m-1)/2)(1-s_{2m})+((2m+k)/2)s_{2m}) \\ &= (1/2)(2m\cdot s_{2m}+((1+k)/2)s_{2m}+(m-1/2)). \end{aligned}$$

Let $C_1=c(P_S)$ and $C_2=c(Q_S)$. Again by assumption (2), we obtain

$$\begin{aligned} C_1-C_2 &= \textstyle\sum_{i=1}^{m}(s_{2i-1}-s_{2i}) + \sum_{i=m+1}^{2m}(s_{2i-1}-s_{2i}) + (s_{4m+1}-s_{4m+2}) + s_{4m+3} \\ &= (1-s_{2m})/2 - (s_{2m}/(2m+k+1))m + (s_{2m}/(2m+k+1))\mu \\ &= 1/2 - s_{2m}(1-(2\mu+k+1)/(2(2m+k+1))), \end{aligned}$$

where $\mu=0$ if $k=0$, $\mu=1$ if $k=1$ or 2, and $\mu=2$ if $k=3$.

Letting $\delta x\to 0$ in $x<s_{2m}<x+\delta x$, we get

$$\begin{aligned} T = (1/2)(C_1+C_2) &= (1/2)((1+k)/2)x + (m-1/2) \\ &= ((2m-1)/4)(1+x(2+(3+k)/(2m-1))), \end{aligned}$$

and $\quad C_S = (1/2)(C_1+C_2+|C_1-C_2|) = T + (1/4)|1-2x(1-(2\mu+k+1)/(2(2m+k+1)))|$.

The performance of the algorithm will be expressed by C_S/T instead of C_S/C^*, since C^* is hard to obtain, and $C_S/C^* \leq C_S/T$ because $C^* \geq T$.

The expected value of C_S/T is given by

$$\begin{aligned} E(C_S/T) &= \textstyle\int_0^1 (1+(1/(2m-1))|1-2x(1-(2\mu+k+1)/(2(2m+k+1)))|/(1+x(2+(3+k) \\ &\qquad /(2m-1)))\cdot((4m+k)!/((2m-1)!(2m+k)!))x^{2m+k}(1-x)^{2m-1}\ dx \\ &= 1 + ((4m+k)!/((2m-1)!(2m+k)!))\cdot 1/(2m-1) \textstyle\int_0^1 |1-2x(1-(2\mu+k+1) \\ &\qquad /(2(2m+k+1)))|/(1+x(2+(3+k)/(2m-1))) x^{2m+k}(1-x)^{2m-1}\ dx, \end{aligned}$$

since $(r+t)!/((r-1)!t!) \int_0^1 x^t(1-x)^{r-1}\ dx = 1$.

We call $E(C_S/T)-1$ the expected relative error. Now we are ready to prove the following.

Theorem 1. $E(C_S/T)-1 = \alpha(m,k)(1/(2m-1))$, where $0<\alpha(m,k)<1$.

Proof Let $f(x;m,k) = |1-2x(1-(2\mu+k+1)/(2(2m+k+1)))|/(1+x(2+(3+k)/(2m-k)))$.

Then
$$\begin{aligned} E(C_S/T) &= 1 + \frac{(4m+k)!}{(2m-1)!(2m+k)!} \frac{1}{2m-1} \int_0^1 f(x;m,k)\, x^{2m+k}(1-x)^{2m-1}\, dx \\ &= 1 + \frac{(4m+k)!}{(2m-1)!(2m+k)!} \frac{1}{2m-1} f(\xi;m,k) \int_0^1 x^{2m+k}(1-x)^{2m-1}\, dx \\ &= 1 + \alpha(m,k)(1/(2m-1)), \end{aligned}$$
where $0<\xi<1$ and hence $0<\alpha(m,k)=f(\xi;m,k)<1$. ☐

It is easy to see that (m,k) is small. In fact, as shown in Theorem 2 $\alpha(m,k)$ is proportional to $m^{-1/2}$ for large m.

Asymptotic analysis

We now analyze the expected behavior of the algorithm when the size of the problem is large.

For large m,
$$E(C_S/T) \sim 1 + \frac{(4m+k)!}{(2m-1)!\,(2m+k)!} \frac{1}{2m} \int_0^1 \frac{|1-2x|}{1+2x} x^{2m+k}(1-x)^{2m-1}\, dx.$$
We let
$$\begin{aligned} V &= \int_0^1 (1-2x)/(1+2x)\cdot x^{2m+k}(1-x)^{2m-1}\, dx \\ &= \mu \int_0^{1/2} (1-2x)\, x^{2m+k}(1-x)^{2m-1}\, dx + \nu \int_{1/2}^1 (2x-1)\, x^{2m+k}(1-x)^{2m-1} dx \end{aligned}$$
where $\mu=(1+2\xi)^{-1}$ and $\nu=(1+2\zeta)^{-1}$ for some $0\le\xi\le 1/2$ and $1/2\le\zeta\le 1$. Thus, $1/2\le\mu\le 1$ and $1/3\le\nu\le 1/2$. For an integer $u\ge 0$, we obtain the following by integrating by parts u+1 times successively:
$$\int_0^{1/2} x^{2m+u}(1-x)^{2m-1}\, dx = \left(\prod_{i=0}^{u} \frac{2m+u-i}{4m+u-i}\right) \int_0^{1/2} x^{2m-1}(1-x)^{2m-1}\, dx - (1/2)^{4m+u} \sum_{i=0}^{u} \frac{2^i}{4m+u-i} \left(\prod_{j=0}^{i-1} \frac{2m+u-j}{4m+u-j}\right),$$
and
$$\int_{1/2}^1 x^{2m+u}(1-x)^{2m-1}\, dx = \left(\prod_{i=0}^{u} \frac{2m+u-i}{4m+u-i}\right) \int_{1/2}^1 x^{2m-1}(1-x)^{2m-1}\, dx - (1/2)^{4m+u} \sum_{i=1}^{u} \frac{2^i}{4m+u-i} \left(\prod_{j=0}^{i-1} \frac{2m+u-j}{4m+u-j}\right),$$
where $\prod_{j=0}^{t} (\ ) = 1$ if $t<0$.

Also, by using Stirling's approximation formula, we get
$$(1/2)^{4m-1} \sim \sqrt{2m/\pi}\,(2m-1)!(2m-1)!/(4m-1)!.$$
Thus,
$$\begin{aligned} &\int_0^{1/2} (1-x)\, x^{2m+k}(1-x)^{2m-1}\, dx \\ &= \int_0^{1/2} x^{2m+k}(1-x)^{2m-1}\, dx - 2\int_0^{1/2} x^{2m+k+1}(1-x)^{2m-1}\, dx \\ &= \left(\prod_{i=0}^{k} \frac{2m+k-i}{4m+k-i} - 2\left(\prod_{i=0}^{k+1} \frac{2m+k+1-i}{4m+k+1-i}\right)\right) \int_0^{1/2} x^{2m-1}(1-x)^{2m-1}\, dx \\ &\quad -(1/2)^{4m+k}\left(\sum_{i=0}^{k} \left(\frac{2^i}{4m+k-i} \prod_{j=0}^{i-1} \frac{2m+k-j}{4m+k-j}\right) - \sum_{i=0}^{k+1} \left(\frac{2^i}{4m+k+1-i} \prod_{j=0}^{i-1} \frac{2m+k+1-j}{4m+k+1-j}\right)\right) \\ &= -\frac{k+1}{4m+k+1}\left(\prod_{i=0}^{k} \frac{2m+k-i}{4m+k-i}\right) \frac{(2m-1)!(2m-1)!}{(4m-1)!} \end{aligned}$$

$$+ (1/2)^{4m-1}(1/2)^{k+1} \left(\frac{2^{k+1}}{4m} \prod_{j=0}^{k} \frac{2m+k+1-j}{4m+k+1-j} + O(m^{-2}) \right)$$

$$\sim (2m-1)!(2m-1)!/(4m-1)! \cdot (1/2)^{k+3} \cdot (1/\sqrt{m}) \cdot (\sqrt{2/\pi} - (k+1)/\sqrt{m})$$

$$\sim (2m-1)!(2m-1)!/(4m-1)! \cdot (1/2)^{k+3} (\sqrt{2/(\pi m)}).$$

Similarly,

$$\int_{1/2}^{1} (2x-1)\, x^{2m+k}(1-x)^{2m-1}\, dx \sim \frac{(2m-1)!(2m-1)!}{(4m-1)!} (1/2)^{k+3}(\sqrt{2/(\pi m)}).$$

Therefore,

$$V \sim (\mu+\nu)\, (1/2)^{k+1} \frac{(2m-1)!(2m-1)!}{(4m-1)!} (\sqrt{2/(\pi m)}).$$

We have the following result.

Theorem 2. For the instances of large problem size,

$$E(C_S/T)-1 \sim \beta\, m^{-3/2}, \text{ where } .08 \leq \beta \leq .15 .$$

Proof Substitution of the value of V above into the expression for $E(C_S/T)$ gives

$$E(C_S/T) \sim 1 + \frac{(4m+k)!}{(2m-1)!(2m+k)!} \frac{1}{2m} (\mu+\nu)(1/2)^{k+3} \frac{(2m-1)!(2m-1)!}{(4m-1)!} \frac{\sqrt{2}}{\sqrt{\pi}} \frac{1}{\sqrt{m}}$$

$$\sim 1 + (\mu+\nu)(1/(4\sqrt{2\pi}))\, m^{-3/2}.$$

Since $5/6 \leq \mu+\nu \leq 3/2$, the theorem is proved. ☐

LPT SCHEDULING ALGORITHM

The LPT(largest-processing-time-first) algorithm is given below:

```
1. Sort S in nondecreasing order such that $s_1 \leq s_2 \leq \dots \leq s_n$.
2. For i=n by -1 to 1 do
       if $c(P_L) \leq c(Q_L)$ then insert $s_i$ into $P_L$
                               else insert $s_i$ into $Q_L$.
```

We note that it was shown in [3] that $C_L/C^* \leq 7/6$. This bound is approachable for any size of the problem. To analyze the LPT algorithm, we identify the last element which is inserted into the partition of greater total sum. Let s_{r+1} be the element such that $c(P_L-\{s_{r+1}\}) \leq c(Q_L-\{s_1,\dots, s_r\})$ and $c(P_L) \geq c(Q_L)$.

In addition to assumption (1), we make the following:

The set $\{s_1,s_2,\dots,s_r\}$ and $\{s_{r+2},s_{r+3},\dots,s_n\}$ are from uniformly distributed sample space over $(0,s_{r+1})$ and $(s_{r+1},1)$, respectively. (3)

Let $x = s_{r+1}$, $f = \sum_{i=i+2}^{n} s_i$, $g = \sum_{i=1}^{r} s_i$ and $w = c(Q_L-\{s_1,s_2,\dots,s_r\})-c(P_L-\{s_{r+1}\})$. We shall assume that

w is uniformly distributed over the interval (0,x). (4)

Then

$$c(P_L) = (1/2)f + x,$$

$$c(Q_L) = (1/2)f + g,$$

$$T = (1/2)(f+x+g), \text{ and}$$

$$C_L/T = ((1/2)f+x)/((1/2)(f+x+g)) = 1 + (x-g)/(f+x+g)$$

Let $p(r,x)$ be the probability that $x < s_{r+1} < x+\delta x$ for some x in $(0,1)$. Then by assumption (1),

$$p(r,x) = (n!/((n-r-1)!r!))\ x^r(1-(x+\delta x))^{n-r-1}\ \delta x + O(\delta x^2).$$

For a given r, to find the average of C_L/T, we need, for a given x, the probability that $(x-w)-g \geq 0$ and the expected value of $(x-w)-g$ for those cases. Let $z(r,x-w)$ be the probability that $(x-w)-(s_1+s_2+\ldots+s_r) \geq 0$ and $q(r,x-w)$ be the expected value of $(x-w)-(s_1+s_2+\ldots+s_r)$ for the cases in which $(x-w)-(s_1+s_2+\ldots+s_r) \geq 0$. We derive expressions for $z(r,x-w)$ and $q(r,x-w)$. Let $\{x_1,x_2,\ldots,x_r\}$ be the set of variables for $\{s_1,s_2,\ldots,s_r\}$ such that for each j, $1\leq j\leq r$, x_j is the variable for some s_i, $1\leq i\leq r$. Let $h(t) = \sum_{i=1}^{t} x_i$ and $x' = x-w$.

To satisfy $x'-(x_1+x_2+\ldots+x_r) \geq 0$ the following must each be satisfied:

$$0 \leq x_1 \leq x',$$

$$0 \leq x_2 \leq x'-x_1 = x'-h(1),$$

$$\vdots$$

$$0 \leq x_r \leq x'-(x_1+x_2+\ldots+x_{r-1}) = x'-h(r-1).$$

Thus $z(r,x')$, the probability that $x'-(x_1+x_2+\ldots+x_r) \geq 0$ when $0<x_i<1$ for $1\leq i\leq r$, is given by

$$z(r,x') = \frac{\int_0^{x'} \int_0^{x'-h(1)} \cdots \int_0^{x'-h(r-1)} dx_r \ldots dx_2\, dx_1}{\int_0^1 \int_0^1 \cdots \int_0^1 dx_r \ldots dx_2\, dx_1}$$

$$= a(r,x')/1 = a(r,x')$$

Similarly $q(r,x')$, the expected value of $x'-g = x'-h(r)$ over $\{(x_1,x_2,\ldots,x_r) \mid x'-h(r) \geq 0\}$ is given by

$$q(r,x') = \frac{1}{a(r,x')} \int_0^{x'} \int_0^{x'-h(1)} \cdots \int_0^{x'-h(r-1)} (x'-h(r-1))\ dx_r \ldots dx_1$$

$$= b(r,x')/a(r,x').$$

Lemma 1. $a(r,x') = x'^r/r!$ for $r\geq 0$.

Proof By induction on r. Obviously, $a(0,x') = 1$. $a(1,x') = \int_0^{x'} dx_1 = x'$. Let $y = x'-x_1$ and $y_i = x_{i+1}$ for $1\leq i\leq r$. Then

$$a(r+1,x') = \int_0^{x'} a(r,y)\ dx_1 = \int_0^{x'} (y^r/r!)\ dx_1 = \int_0^{x'} ((x'-x_1)^r/r!)\ dx_1$$

$$= x'^{r+1}/(r+1)!.$$ □

Lemma 2. $b(r,x') = x'^{r+1}/(r+1)!$ for $r\geq 0$.

Proof A proof by induction on r similar to that of Lemma 1 works. □

Thus we have $z(r,x') = x'^r/r!$ and $q(r,x') = x'/(r+1)$.

Theorem 3. The expected performance ratio for LPT algorithm is given by

$$E(C_L/T)-1 \sim \gamma\ n^{-2}, \text{ where } \gamma \text{ is a constant } .50\leq\gamma\leq 2.72.$$

Proof $E(C_L/T)-1 \geq \int_0^1 p(0,x)(\int_0^x z(0,x-w)\ 2q(0,x-w)/(n(1+x))\ dw/\int_0^x dw)\ dx$

$$= \int_0^1 ((n!(1-x)^{n-1}/(n-1)!)\ x^{-1} \int_0^x (2(x-w)/(n(1+x))\ dw)\ dx$$

$$= \int_0^1 (1+x)^{-1}(1-x)^{n-1}x\ dx \geq (1/2) \int_0^1 (1-x)^{n-1}x\ dx = \frac{1}{2n(n+1)}.$$

$$E(C_L/T)-1 \leq \sum_{r=0}^{n-1} \int_0^1 p(r,x)\ (\int_0^x z(r,x-w)\ \frac{2q(r,x-w)}{(n-r)(1+x)} dw \ / \int_0^x dw)\ dx$$

$$= \sum_{r=0}^{n-1} \int_0^1 \frac{n!}{(n-r-1)!r!}\ x^r(1-x)^{n-r-1}\ x^{-1}$$

$$(\int_0^1 \frac{(x-w)^r}{r!x^r}\ \frac{2(x-w)}{(n-r)(1+x)(r+1)}\ dw)\ dx$$

$$= \sum_{r=0}^{n-1} \int_0^1 \frac{2n!}{(n-r)!r!}\ \frac{(1-x)^{n-r-1}}{x(1+x)}\ (\int_0^x \frac{(x-w)^{r+1}}{(r+1)!}\ dw)\ dx$$

$$= \sum_{r=0}^{n-1} \int_0^1 \frac{2n!}{(n-r)!r!}\ \frac{(1-x)^{n-r-1}}{x(1+x)}\ \frac{x^{r+2}}{(r+2)!}\ dx$$

$$= \sum_{r=0}^{n-1} 2n!/((n-r)!(r+2)!r!) \int_0^1 x^{r+1}(1-x)^{n-r-1}\ dx$$

$$= \sum_{r=0}^{n-1} (2n!/((n-r)!(r+2)!r!))((r+1)!(n-r-1)!/(n+1)!)$$

$$= \sum_{r=0}^{n-1} 2/((n+1)(n-r)(r+2)r!) \leq n^{-2} \sum_{r=0}^{n-1} 1/r!$$

$$\leq n^{-2} \sum_{r=0}^{\infty} 1/r! = n^{-2}\ e. \quad \square$$

COMPUTATIONAL EXPERIMENTS

Computational experiments were performed to confirm the validity of the analysis of the expected performance. The experiments involved random samples of size 20, 40, 80, 160, 320, and 640. For each n, 100 samples were generated. For each sample, C, T and C/T were computed. Then $(C/T)_{avg}$, the average of C/T over 100 samples, was obtained, and from this, the coefficient, either β or γ, was computed. These values appear in the table.

n	$(C_S/T)_{avg}$	β_{avg}	$(C_L/T)_{avg}$	γ_{avg}
20	1.1150	0.13	1.00366	1.47
40	1.00474	0.15	1.00096	1.54
80	1.00152	0.14	1.00021	1.38
160	1.00055	0.14	1.00006	1.56
320	1.00018	0.13	1.00002	1.53
640	1.00006	0.13	1.00000	1.51

The results of the experiments seem to confirm the validity of the analysis that the expected relative errors are proportional to $n^{-3/2}$ and n^{-2}.

ε-APPROXIMATION ALGORITHM

In this section we present an ε-approximation algorithm. It finds for any $0<\varepsilon<1$ an approximate solution C_ε satisfying $C_\varepsilon/C^* < 1+\varepsilon$ and has time and space complexity of $O(n)+O(\varepsilon^{-2})$. The dynamic programming portion of the algorithm requires time and space $O(\varepsilon^{-2})$ which is dependent only on ε and independent of the problem size n.

A simple routine ASSIGN which will be used as a subroutine in the algorithm is given.

```
ASSIGN(T',i,P')
1. Find the largest index k such that $s_i+s_{i+1}+\ldots+s_k \leq T'$.
2. If $T'-(s_i+\ldots+s_k) \leq (s_i+\ldots+s_k+s_{k+1})-T'$
     then set $P' \leftarrow \{i,\ldots,k\}$
     else set $P' \leftarrow \{i,\ldots,k,k+1\}$.
```

The following is the notation used in the algorithm:

(a) ε is a real number, $0<\varepsilon<1$.

(b) T is the sum of all the elements in S divided by 2, $T = c(S)/2$.

(c) SMALL is a list of indices i of elements such that $s_i<T\varepsilon$.

(d) LARGE is a list of indices i of elements such that $s_i\geq T\varepsilon$.

(e) TABLE is an array of length $\lfloor 1/\varepsilon \rfloor+1$. TABLE(k) = $\emptyset$ or is of the form <LL(k),L(k),LU(k),U(k)>. LL(k) is a list of indices $i_1,\ldots,i_t$ with $L(k) = s_{i_1}+\ldots+s_{i_t}$ and $k = \lfloor L(k) \rfloor$, and LU(k) is a list of indices $j_1,\ldots,j_r$ with $U(k) = s_{j_1}+\ldots+s_{j_r}$ and $k = \lfloor U(k) \rfloor$. Also $L(k) \leq U(k)$.

The ε-approximation algorithm is described below:

```
$\varepsilon$-PARTITION($S,n,P_\varepsilon,Q_\varepsilon$)
1. Find c(S), the sum of all the elements in S, and set $T \leftarrow c(S)/2$.
2. [Separate elements in S into two groups.]
   For i=1 to n do
      if $s_i/T \geq \varepsilon$ then put $s_i$ into LARGE
                  else put $s_i$ into SMALL.
   Let LARGE = $\{s_1,\ldots,s_m\}$ and SMALL = $\{s_{m+1},\ldots,s_n\}$.
3. If $c(\mathrm{SMALL}) \geq T$ then call ASSIGN($T,m+1,P_\varepsilon$) stop.
4. [Initialize the dynamic programming part.]
   Set $g \leftarrow \lfloor (T-c(\mathrm{SMALL}))/T\varepsilon \rfloor$; $h \leftarrow \lfloor 1/\varepsilon \rfloor$; TABLE(0) $\leftarrow \langle\emptyset,0,\emptyset,0\rangle$ and
      TABLE(j) $\leftarrow \emptyset$ for all $1\leq j\leq g$.
5. [Update TABLE.]
   For i=1 to m do
      for j=g-1 by -1 to 0 do
         if TABLE(j) $\neq \emptyset$
            then set $f_L \leftarrow \lfloor (L(j)+s_i)/(T\varepsilon) \rfloor$ and $f_U \leftarrow \lfloor (L(j)+s_i)/(T\varepsilon) \rfloor$
                 if $g \leq f_U \leq h$
5.1                 then call ASSIGN($T-(U(j)+s_i),m+1,P'$);
                         $P_\varepsilon \leftarrow LU(j) \cup \{i\} \cup P'$ stop
            if $g \le f_L \le h$
5.2           then call ASSIGN(T-(L(j)+$s_i$),m+1,P');
                   $P_\varepsilon \leftarrow$ LL(j) $\cup \{i\} \cup$ P' stop
            if $f_U < g$
5.3           then if TABLE($f_U$) = $\emptyset$
                     then TABLE($f_U$) $\leftarrow$ {LU(j) $\cup$ {i}, U(j)+$s_i$,
                                          LU(j) $\cup$ {i}, U(j)+$s_i$}
                   if U(j)+$s_i$ < L($f_U$)
                     then LL($f_U$) $\leftarrow$ LU(j) $\cup$ {i}; L($f_U$) $\leftarrow$ U(j)+$s_i$
                   if U(j)+$s_i$ > U($f_U$)
                     then LU($f_U$) $\leftarrow$ LU(j) $\cup$ {i}; U($f_U$) $\leftarrow$ U(j)+$s_i$
            if $f_L < g$
5.4           then if TABLE($f_L$) = $\emptyset$
                     then TABLE($f_L$) $\leftarrow$ {LL(j) $\cup$ {i}, L(j)+$s_i$,
                                          LL(j) $\cup$ {i}, L(j)+$s_i$}
                   if L(j)+$s_i$ < L($f_L$)
                     then LL($f_L$) $\leftarrow$ LL(j) $\cup$ {i}; L($f_L$) $\leftarrow$ L(j)+$s_i$
                   if L(j)+$s_i$ > U($f_L$)
                     then LU($f_L$) $\leftarrow$ LL(j) $\cup$ {i}; U($f_L$) $\leftarrow$ L(j)+$s_i$
6. Find the largest index k such that TABLE(k) $\ne \emptyset$;
   set $P_\varepsilon \leftarrow$ LU(k) $\cup$ SMALL stop.
```

We need a lemma before we prove the result of this section.

Lemma 3. Let $f_t = \lfloor (s_{i_1}+\ldots+s_{i_t})/T\varepsilon \rfloor$, where $\{s_{i_1},\ldots,s_{i_t}\}$ is a subset of LARGE and $i_1 < \ldots < i_t$. Then after the execution of step 5 for $i = i_1,\ldots,i_t$, TABLE$(f_t) \ne \emptyset$, and $L(f_t) \le s_{i_1}+\ldots+s_{i_t}$ and $U(f_t) \ge s_{i_1}+\ldots+s_{i_t}$.

Proof By induction on t. Consider the execution of step 5 for $i = i_1$. We note that TABLE(0) $\ne \emptyset$. If TABLE$(f_1) = \emptyset$, then TABLE(f_1) will be updated to $<\{i_1\}, s_{i_1}, \{i_1\}, s_{i_1}>$. Suppose TABLE$(f_1) \ne \emptyset$. If $L(f_1) > s_{i_1}$, then $LL(f_1)$ and $L(f_1)$ are updated to $\{i_1\}$ and s_{i_1}, and if $U(f_1) < s_{i_1}$, then $LU(f_1)$ and $U(f_1)$ are updated to $\{i_1\}$ and s_{i_1}. Thus, TABLE$(f_1) \ne \emptyset$ and $L(f_1) \le s_{i_1} \le U(f_1)$. Suppose that the assertions of the lemma are true for $t = k$, and consider what happens when step 5 is executed for $i = i_{k+1}$. TABLE$(f_k) \ne \emptyset$ and $L(f_k) \le s_{i_1}+\ldots+s_{i_k} \le U(f_k)$ by hypothesis and $U(f_k)-L(f_k) < T\varepsilon$. Therefore $\lfloor (L(f_k)+s_{i_{k+1}})/T\varepsilon \rfloor - \lfloor (U(f_k)+s_{i_{k+1}})/T\varepsilon \rfloor \le 1$. Thus $f_{k+1} = \lfloor (L(f_k)+s_{i_{k+1}})/T\varepsilon \rfloor$ or $f_{k+1} = \lfloor (U(f_k)+s_{i_{k+1}})/T\varepsilon \rfloor$, and so TABLE$(f_{k+1}) \ne \emptyset$ and $L(f_t) \le s_{i_1}+\ldots+s_{i_{k+1}}$ and $U(f_t) \ge s_{i_1}+\ldots+s_{i_{k+1}}$. □

Theorem 4. The algorithm has time and space complexity $O(n)+O(\varepsilon^{-2})$ and produces an approximate solution C_ε satisfying $C_\varepsilon/C^* \leq 1+\varepsilon$ for any $0<\varepsilon<1$.

Proof Since LL(k) and LU(k) contain no more than $O(\varepsilon^{-1})$ elements, the size of TABLE is $O(\varepsilon^{-2})$. The input takes $O(n)$ space. Therefore, the space complexity is $O(n)+O(\varepsilon^{-2})$. For the time complexity, ASSIGN is executed only once taking $O(n)$ time. Step 2 takes $O(n)$ time. Since m, the number of elements in LARGE, is less than $2\lceil\varepsilon^{-1}\rceil$, steps 5.1-5.4 are executed $O(\varepsilon^{-1})$ times, each requiring $O(\varepsilon^{-1})$ time. Thus the time complexity of the algorithmis $O(n)+O(\varepsilon^{-2})$.

For the error analysis, we consider the following three cases.

Case 1: The algorithm terminates in step 3. Since $s_i < T$ for all $m+1<i\leq n$, $C_\varepsilon < T+T\varepsilon = (1+\varepsilon)T$. Obviously $C^* \geq T$. Hence $C_\varepsilon/C^* < 1+\varepsilon$.

Case 2: The algorithm terminates in step 5.3. Then $T-c(\text{SMALL}) \leq L(j)+s_i \leq T$. Thus, $|c(P_\varepsilon)-c(Q_\varepsilon)| < T\varepsilon$, and so $C_\varepsilon/C^* < 1+\varepsilon$.

Case 3: The algorithm terminates in step 6. Let $P^* = P_L \cup P_S$ and $Q^* = Q_L \cup Q_S$ be an optimal partition of S, where P_L, $Q_L \subseteq$ LARGE and P_S, $Q_S \subseteq$ SMALL. Assume that $c(P_L) \leq c(Q_L)$. Then $c(P_L) < T-c(\text{SMALL})$, since otherwise the algorithm must terminate in step 5 while examining the elements in P_L by Lemma 3. Therefore $Q_S = \emptyset$, since $C^* = c(Q^*) = c(Q_L \cup Q_S) \geq c(Q_L)$. Let $f' = \lfloor c(P_L)/T\varepsilon \rfloor$. Then by Lemma 3, $\text{TABLE}(f') \neq \emptyset$ and $U(f') \geq c(P_L)$ and $U(f')+c(\text{SMALL}) \geq c(P_L)+c(\text{SMALL})$. Thus $C_\varepsilon = C^*$.

Acknowledgement The work on LPT algorithm was done jointly by Greg N. Frederickson and the author.

References

1. Cook, S. A., "The Complexity of Theorem Proving Procedures," Proc. STOC (1971), 151-158.
2. Garey, M. R. and Johnson, D. S., "Approximation Algorithms for combinatorial Problems: An Anntated Bibliograph," in Algorithms and complexity, J. F. Traub, ed., Academic Press(1976), 41-52.
3. Graham, R. L., "Bounds on Multiprocessing Time Anomalies," SIAM J. Appl. Math. 17(1969), 416-429.
4. Grimmet, G. R. and McDiarmid, C. J. H., "On Coloring Random Graphs," Math. Proc. Camb. Phil. Soc. 77(1975), 313-324.
5. Ibarra, O. H. and Kim, C. E., "Fast Approximation Algorithms for the Knapsack and Sum of Subset Problems," JACM 22(1975), 463-468.
6. Johnson, D. S., "Approximation Algorithms for Combinatorial Problems," JCSS 9(1974), 256-278.
7. Karp, R. M., "Reducibility among Combinatorial Problems," in Complexity of Computer Computations, R. E. Miller and J. W. Thatcher, eds., Plenum Press, N.Y.(1972), 85-104.
8. Karp, R. M., "The Probabilistic Analysis of Some Combinatorial Search Algorithms," in Algorithms and Complexity, J. F. Traub, ed., Academic Press(1976), 1-19.
9. Rosenkrantz, D. J., Stearns, R. E. and Lewis, P. M., "Approximate Algorithms for the Traveling Salesperson Problem," Proc. SWAT(1974), 33-

42.

10. Sahni, S. K., "Approximate Algorithms for the 0/1 Knapsack Problem," JACM 22(1975), 115-124.

AN INFORMATION STRUCTURE FOR POSETS*

E. LODI and F. LUCCIO
University of Pisa, Pisa, Italy

SUMMARY

Partial ordering relations are often defined on the elements of a set of data, however, little attention has been devoted to partly ordered sets (posets) by a computational point of view.

A basic problem is to discover an information structure to hold a poset, such that the most frequent operations on the data can be efficiently performed. Digraphs associated with the partial ordering relation (in particular, the Hasse diagram) seem to be the only structures commonly in use.

Starting from Dilworth's theorem on chain decomposition, we study the complexity of finding different sets of chains that cover the poset, and propose information structures to hold such sets of chains. These structures are composed of interconnected balanced trees (IBT). Each tree of a IBT holds a chain of the poset, and the "immediate precedence" relation between elements of different chains is represented by pointers between trees.

The use of IBT is mainly intended to increase the efficiency of the operations of search for an element and insertion of a new element. In the latter case (i.e., for dinamically variant structures) updating the pointers between trees becomes a serious problem, since the pairs of elements connected by an immediate precedence relation may substantially change. To ease this phase of insertion, a special variant of the binary search traversal is introduced.

---

* This research is supported in part by Comitato Nazionale per le Scienze Matematiche, C.N.R., under contract No. 75.01035.01.

THE FORD-JOHNSON SORTING ALGORITHM IS NOT OPTIMAL (EXTENDED ABSTRACT)

GLENN K. MANACHER
Department of Information Engineering and
Computer Center
University of Illinois, Chicago Circle
Chicago, Illinois 60680

Let $S(t)$ denote the minimum number of comparisons necessary to sort t items. Is there an algorithm that will sort t items in $S(t)$ comparisons for all t? A contender since 1959 for this honor has been the "merge-insertion" algorithm due to L. R. Ford and S. B. Johnson. While few have conjectured that it is optimal, there is no known t for which $S(t) < F(t)$, where $F(t)$ is the number of comparisons required by the Ford-Johnson algorithm [FJA] .

In this paper, we produce by construction an infinite number of values of t for which $S(t) < F(t)$. The smallest we have found is $t = 189$. The main theorem states that these values of t come in "bands"; in particular, if $u_k \equiv \lfloor (4/3)2^k \rfloor$, then for all $k \geq 7$ there exists an integer $b(k) = O(\log u_k)$ such that for all m in the region $b(k) \leq m \leq u_{k-3}$, $t = u_k + m$ elements may be sorted in $G(t) < F(t)$ comparisons. The method consists simply in first sorting u_k, then m, elements, using the FJA, then merging the sorted lists using a merging algorithm asymptotically slightly faster than the Hwang-Lin merging algorithm. The method for performing the merge is via a new algorithm involving amoung other things the merging of 3 elements with n for small values of n, for which near-optimal techniques are known.

INTRODUCTION

The number of comparisons required by the FJA is [1,2,3]

$$F(t) = \sum_{0 \leq k < t} \lceil \log (3/4)k \rceil \qquad (1)^{*}$$

The details of the FJA are by now well know [2,3] and the reader's attention is directed to these sources.

A sketch of the FJA will nevertheless be helpful. Suppose t is even. Divide the elements into two parts, arbitrarily, so that they have labels $a_1, a_2, \ldots, a_{t/2}; b_1, b_2, \ldots, b_{t/2}$.

---

* All logarithms are to the base 2.

(If t is odd, then the **extra** element becomes $b_{t/2 + 1}$.) Perform t/2 comparisons between each a_i, b_i, then relabel where necessary to insure that $b_i \le a_i$ for $1 \le i \le t/2$.

Now sort the a_i using the FJA recursively on t/2 elements. This requires F(t/2) comparisons. Finally, arrange the b_i into sequences B_j, j = 0, 1, 2, ..., where $B_0 = (b_1)$, $B_1 = \emptyset$, $B_2 = (b_2, b_3)$, $B_3 = (b_4, b_5)$, $B_4 = (b_6, b_7, b_8, b_9, b_{10}$ $b_{11})$, ... Merge each element of the b's into the sorted list (initially just a's, later a's and the b's that get merged in) as follows: First merge b_1 in zero comparisons (since it is already merged in.) Then merge in b_3 by binary insertion in 2 comparisons, then b_2 ditto in 2 comparisons. Then merge b_5 in 3 comparisons, then b_4 in 3 comparisons, then $b_{11}, b_{10}, \ldots, b_4$ in 4 comparisons, always in the order given. Analysis shows easily that (a) the elements of B_j will be inserted in j comparisons, and (b) if $y_i = |B_i|$, then y_i may be given recursively by

$$y_0 = 1$$

$$y_i + 2 \sum_{0 \le j < i} y_j = 2^i \text{ for } i \ge 1 \qquad (2)$$

and this is shown in [2] and [3] to lead to (1).

It is clear that (a) this method depends heavily on insertion of single elements, and that (b) when $t = \sum_{0 \le i \le j} y_i$, "best" results are obtained, because the final j^{th} "piece" of the t elements is merged optimally since it is of length B_j. Therefore we expect some sort of periodic behavior to manifest itself when we express F(t) as a function of log t. This periodic behavior, as shown in[2,3] is exposed by first transforming (1) into

$$F(t) = t \lceil \log (3/4)t \rceil - \lfloor 2^{\lfloor \log 6t \rfloor} / 3 \rfloor + \lfloor \tfrac{1}{2} \log 6t \rfloor \qquad (3)$$

then making the substitution $t = (4/3)\, 2^{k + f}$, where k is an integer and f a real number of the range $0 < f < 1$.

There results

$$F(t) = t \log t - t(1 + f + 2^{1-f} - \log 3) + \tfrac{1}{2} \log t + O(1) \quad (4)$$

in which the periodicity mentioned above can be seen by plotting $c(t) = 1 + f + 2^{1-f} - \log 3$ versus $\log t$.

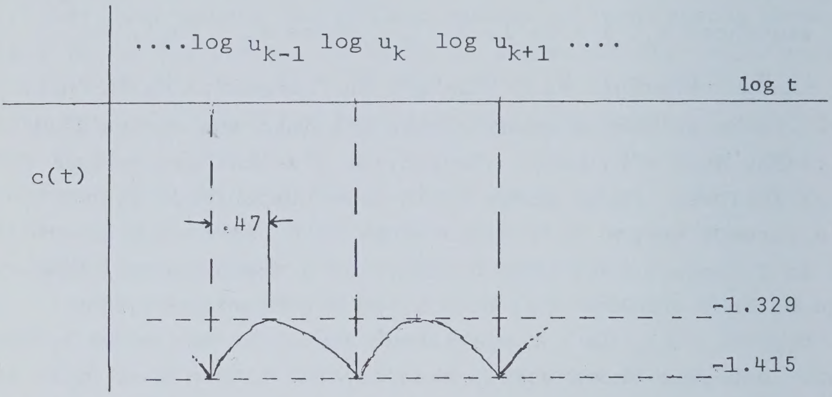


Figure 1

It can be calculated that c(t) varies between $-1.415 = -(3 - \log 3)$ when f = 0 (or 1), and $-1.329 = -(2 - \log 3 + \log e - \log \log e)$ when $f \simeq .47$.

The values of u_k are $\lfloor (4/3)\, 2^k \rfloor$; these are the "best" value. We shall say that $t = u_k$ corresponds to a "cusp" because of the discontinuous behavior of the derivative of c(t) in figure 1.

THE METHOD

An obvious thing to do in an attempt to better the number of comparisons performed by the FJA is to consider $t = u_k + u_{k'}$, sort the u_k and $u_{k'}$ with the FJA, then merge the sorted lists. If $k \neq k'$, the resultant t will have a slight advantage, since c(t) will not be at a cusp.

Set $k' = k - d$. Since $u_k = \lfloor u_{k-1} / 2 \rfloor$, as can easily be shown, it follows that for $k \gg d$, to an excellent approximation, $u_k \doteq 2^d * u_{k-d}$. Define $\alpha = u_k / u_{k-d}$. Then $\alpha \doteq 2^d$. From here on, we will take this approximation to be exact, because we will be working to an order well above the inaccuracy of the approximation.

After about a page of algebra, we find that

$$F(u_k + u_{k-d}) - (F(u_k) + F(u_{k-d}))$$
$$= t((d + 2) / (\alpha + 1)) - \tfrac{1}{2} \log t - 0(1) \qquad (5)$$

where $t = u_k + u_{k-d}$.

The difference, reflected on the right-hand side of (5) mainly by the term linear in t, $t(d + 2) / (\alpha + 1))$ is what must be made up for by the merge.

The best known general-purpose efficient merging algorithm is the Hwang-Lin merging algorithm [4]. The algorithm is well explained in Knuth [2]. We deal here with a slightly simpler version yielding the same worst-case performance. τ is defined to be $\lfloor \log n/m \rfloor$. Both lists will be scanned downward, as in ordinary, "linear" merging. However, in the Hwang-Lin scheme, a block of 2^τ elements at the top of the large list is considered at each iteration (assuming that many elements remain). The topmost element of the small list is compared to the botton element of the block. If it is larger, the block is "annexed" from the large list (i.e., is removed and enters the merged list). If it is smaller (or equal), then the element from the small list is merged into the $2^\tau - 1$ topmost elements of the large list by binary insertion in τ comparisons. This leads [2] to a simple formula for the number of comparisons required to merge m items with n in the worst case:

$$H(m,n) = m(\tau + 1) + \lfloor n^\tau / 2 \rfloor - 1 \qquad (6)$$

For $n = 2^d m$, this reduces to

$$H(m, 2^d m) = m(d + 2) - 1$$
$$= t((d + 2) / (\alpha + 1)) - 1 \qquad (7)$$

From (5) and (7), and careful attention to the sign of the 0(1) term, there arises the key equation

$$F(t) - (F(u_k) + F(u_{k-d}) + H(u_{k-d}, u_k))$$
$$= -\tfrac{1}{2} \log t - 0(1) \qquad (8)$$

From this equation, two propositions follow:

Lemma 1. Sorting u_k elements, then u_{k-d}, by the FJA, then merging the two sorted lists with the Hwang-Lin algorithm [HLA] will never result in fewer comparisons than sorting $u_k + u_{k-d}$ elements with the FJA.

Lemma 2. If a merging algorithm could be found requiring $H'(m,n) = hm - c$ comparisons, where h and c are constants, $m = u_{k-d}$, $n = u_k$, and $h < d + 2$, then for some k_0, the FJA may be surpassed for all $k \geq k_0$.

Sketch of proof for lemma 2: If this were the case, then equation (8) would be replaced with one in which another, positive, term proportional to t were introduced on the right-hand side. □

We now show that this is possible. Consider the case $d = 3$. Replace the HLA by a subtler algorithm that merges pairs and triples from the head of the small list with 8-tuples and 9-tuples from the head of the large list. (In the case $d = 3$, the HLA would only attempt to merge a singleton element from the small list with a 8-tuple from the large list.) In particular, consider a merging algorithm which iterates the following loop until either list is either exhausted or too few elements remain to permit another iteration. Let "element 1,2,3, ..." refer to the first, second, third, ... remaining element on a list.

1. Compare small [1]: large [8]. If greater, annex 8 elements from large. Else
2. Compare small [2]: large [9]. If greater, insert 1 element in small into 7 elements in large in 3 comparisons and annex 9. Else
3. Compare small [3]: large [9]. If greater, insert 2 elements in the small list into 8 elements of the large list in 6 comparisons and annex 9. Else
4. Insert 3 elements in the small list into 8 elements of the large list in 8 comparisons.

Algorithm QM

Algorithm QM uses the known facts that $M(2,8) = 6$ and $M(3,8) = 8$, where $M(m,n)$ is the minimum number of comparisons required to merge m elements with n.

Analysis reveals that algorithm QM performs as inefficiently as possible when the merging proceeds entirely via steps 1 and 3, when m is even, or via steps 1 and 3, and just one instance of step 2 if m is odd. In either case, the number of comparisons performed is easily shown to have the form $hm - c$, where $h = 4\ 15/16$. Since $d + 2 = 5$, the merging algorithm QM must lead to some value of k_0 specified by lemma 2. In fact, for algorithm QM, $k_0 = 9$, corresponding to $t = 767$.

A similar scheme to algorithm QM, involving the comparisons small [3]: large [11] and small [4]: large [11] yields $h = 4\ 7/8$. (Considering small [j]: large [x] for $j \geq 5$ yields only extremely small further improvements in h, which seem not worth analyzing.) When this scheme is analyzed, it yields $k_0 = 8$, corresponding to $t = 383$.

An optimized version of the algorithm mentioned in the last paragraph, in which the exact formulae for $M(1,n)$ and $M(2,n)$ [5] are utilized, together with known values for $M(3,n)$ for small values n, there results an algorithm whose worst-case behavior may be analyzed using dynamic programming techniques. This algorithm yields $k_0 = 7$, corresponding to $t = 191$. We have not discovered any scheme for achieving a lower value of k_0, and we doubt that any reasonable one exists.

GENERALIZATION TO OTHER VALUES OF d

When $d < 3$, the methods above fail to produce any solutions surpassing the FJA.

However, when $d > 3$, the provable fact [5] that $M(2, \lfloor(17/14)2^d\rfloor - 1) = 2d$ and $M(3, \lfloor(31/28)2^d\rfloor - 1) \leq 3d - 1$ (the latter formula being provable from the former) may be used to establish a version of algorithm QM, which we call algorithm QMA, that holds for all $d > 3$.

1. Compare small [1]: large $[2^d]$. If greater, annex 2^d. Else
2. Compare small [2]: large $[\lfloor(17/14)2^d\rfloor]$. If greater insert element 1 into $2^d - 1$ elements in d comparisons, and annex $\lfloor(17/14)2^d\rfloor$. Else
3. Compare small [3]: large $[\lfloor(31/28)2^d\rfloor]$. If greater, insert 2 elements into $\lfloor(17/14)2^d\rfloor - 1$ elements in 2d comparisons and annex $\lfloor(31/28)2^d\rfloor$. Else
4. Insert 3 elements into $\lfloor(31/28)2^d\rfloor - 1$ elements in $3d - 1$ comparisons.

Algorithm QMA

Analysis again shows that steps 1 and 3 of algorithm QMA are "pessimal". There results

Lemma 3. Algorithm QMA, merging m elements with $2^d m$, with

$d > 3$, requires $hm - c$ comparisons, where $h \leq d + 1 + 31/32$; this is an equality for $d = 4$ and a strict inequality for $d > 4$. In fact, we can easily show that for algorithm QMA,

$$\liminf_{d \to \infty} h = d + 1 + 53/56 \tag{9}$$

FURTHER GENERALIZATION

The same analysis that led to (8) is also -- surprisingly -- valid when we drop the requirement that both the large and the small lists be resp. of length u_k and u_{k-d}. Indeed, let $m \in \{u_{k-d-1} + 1, u_{k-d-1} + 2, \ldots, u_{k-d}\}$. Then formula (8) proves to remain true when "u_{k-d}" is replaced by m. This fact, combined with lemma 3, and the results concerning the existence of optimized versions of our algorithms for small values of k_0 (i.e., $k_0 = 7$ and $k_0 = 8$) yields upon analysis our final

Theorem. For all $k \geq 7$, there exists an integer $b(k) = O(\log u_k) = O(k)$ such that for all m in the range $b(k) \leq m \leq u_{d-3}$, $t = u_k + m$ elements may be sorted with $G(t) < F(t)$ comparisons. □

A computer experiment reveals that $b(7) = 19$ and $b(8) = 22$. The former results tells us that $t = 19 + u_7 = 189$ elements may be sorted in fewer comparisons than the FJA requires. This is the smallest value of t for which we have been able to achieve this result.

OPEN QUESTIONS

We do not have any clue about the following questions.

1. Is there a single algorithm that will sort t items in time $S(t)$?
2. Is the periodic behavior of $c(t)$ (as shown in fig. 5) an intrinsic property of $S(t)$, in the sense that waves of some shape exist?
3. Does there exist a real number v, such that for all $t \geq t_0$, $S(t) - \lceil \log t! \rceil > vt$?
4. Our main theorem deals only with values of f (cf. (4)) which lie in the range $0 < f < .17 = \log 9/8$. Are there methods which will beat the FJA for values of t corresponding to values of f outside this range?
5. Are the solutions for the FJA always optimal at the "cusps" $t = u_k$? That is, does $F(u_k) = S(u_k)$ for all k?

REFERENCES

1. Ford, L. R., Jr., and S. B. Johnson. "A Tournament Problem". AMM 66, 5, 1959, pp. 387-389.
2. Knuth, Donald. "The Art of Computer Programming", Vol. III, Addison-Wesley, 1973, pp. 181-209.
3. Hwang, F., and S. Lin. "An Analysis of Ford and Johnson's Sorting Algorithm". Proc. Third Annual Princeton Conf. on Information Science and Systems, 1969, pp. 292-296.
4. Hwang, F. and S. Lin. "A Simple Algorithm for Merging Two Disjoint Linearly-Ordered Sets". *SIAM J. Computation* 1, 1, 1972, pp. 31-39.
5. Graham, R. L. "On Sorting by Comparisons". Proc. of the Second Atlas Conference, 1971.

ON THE AVERAGE NUMBER OF MAXIMA IN A SET OF VECTORS AND APPLICATIONS

J. L. BENTLEY, H. T. KUNG, M. SCHKOLNICK*, C. D. THOMPSON
Carnegie-Mellon University, Pittsburgh, Pennsylvania

ABSTRACT

A maximal vector of a set is one which is not less than any other vector in all components. We derive a recurrence relation for computing the average number of maximal vectors in a set of n vectors in d-space under the assumption that all $(n!)^d$ relative orderings are equally probable. Solving the recurrence shows that the average number of maxima is $O((\ln n)^{d-1})$. We use this result to construct an algorithm for finding all the maxima that has expected running time linear in n (for sets of vectors drawn under our assumptions). The result is also used to derive an upper bound on the expected number of points which are on the boundary of the convex hull of a random point set.

1. INTRODUCTION

The problem of finding all maximal vectors in a set of n d-vectors has recently been studied by Kung, Luccio and Preparata [3] and F. Yao [7]. In this paper we consider the related problem of finding the expected number of maximal elements in a given set. We give a solution to that problem under a very general probability distribution and then apply the answer to the solution of related problems.

A maximal vector is one which is not less than any other vector in all components. More precisely, we say that a vector P dominates the vector Q if P is greater than Q in every component; then a vector is maximal if it is not dominated by any other vector in the set. For example, in $\{(1,2,4), (2,3,1), (3,1,3), (4,4,2)\}$, only $(2,3,1)$ is not maximal. It is helpful to view this problem geometrically when $d = 2$. In that case the vectors can be considered as n points in the plane and a given vector is maximal if and only if there is no point in its first quadrant (above it and to its right).

A probability distribution is implied as we ask for the expected number

---

*IBM Research Laboratory, San Jose, California 95193

This research was supported in part by the National Science Foundation under Grant MCS 75-222-55 and the Office of Naval Research under Contract N00014-76-C-0370, NR 044-422.

of maxima. A mathematically tractable yet reasonable model assumes that for each vector, the magnitude of one component is distributed independently of the magnitude of the other components and, for each component, the magnitudes chosen for each vector are distinct. The second restriction implies that the vectors can be sorted into increasing order on any component, yielding a relative ordering from 1 to n. Thus each set of n d-vectors corresponds to a particular relative ordering for each component, that is, to one of $(n!)^d$ assignments of permutations of $(1,2,3,\ldots,n)$ to the d components. Examples of multivariate statistical distributions with distinct components distributed independently include the multivariate normal and multivariate uniform drawn from a unit hypercube. (Recall that elements drawn independently from any continuous distribution function are distinct with probability one.)

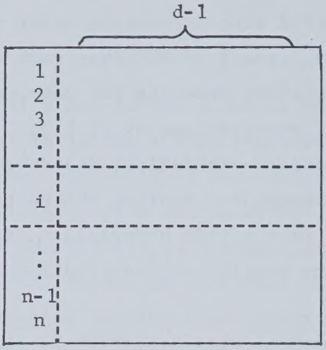
The solution to the maximal vector problem is often required in the analysis of the runtime of dynamic programming algorithms (see Schkolnick [5] and Schkolnick and Thompson [6]). In dynamic programming the solution to a problem of size n is obtained from the best solutions of problems of size n-1. For many applications a cost vector of length one is sufficient, i.e., there is a single best solution to all subproblems. In cases where more than one best solution must be retained for each subproblem, it may still be possible to design a multidimensional cost function with the property that the best solutions for every subproblem are just the maximal ones. If the cost vectors of candidate solutions are assumed to have the proper distribution, then the maximal vector problem indicates the expected number of best solutions.

In Section 2 we formulate and solve a recurrence that shows that the expected number of maxima among n d-vectors is $O((\ln n)^{d-1})$. We use this result in Section 3 to give an algorithm for finding all the maxima of a set of n d-vectors that has expected running time linear in n. In Section 4 we show for a given set of random points, how this result gives an upper bound on the expected number of points from the set which are on the boundary of the convex hull of the set.

2. DETERMINING THE AVERAGE NUMBER OF MAXIMA

In this section we derive the primary result of this paper. The derivation we give is rather informal; a more formal derivation of the result will be given in a longer version of this paper which will appear in the Journal of the ACM.

Let A(n,d) be the average number of maximal vectors in a set of n d-vectors. Without loss of generality, assume that the vector components in each dimension are the integers from 1 to n. To compute the expected number of maxima in a set we will consider the set sorted in order by the first coordinate. The situation we now have is illustrated in the following figure.



We now ask what is the probability that the i-th vector in the set is a maximum? Since its first coordinate is greater than the first coordinates of the 1-st through the (i-1)-st vectors, it cannot be dominated by any of those. Therefore the i-th vector is a maxima if and only if its remaining d-1 coordinates are maximal in the set of the i-th through the n-th vectors. The probability that the i-th vector is a maximal in this set is, by independence, the expected number of maxima in the set (which is A(n-i+1,d-1)) divded by the total number of vectors in the set (which is n-i+1). Since these probabilities are independent for all values of i, to find the expected number of maxima in the set we sum the probabilities of each vector being maximal and we have

$$A(n,d) = \sum_{i=1}^{n} \frac{A(n-i+1,d-1)}{n-i+1}$$

$$= \sum_{j=1}^{n} \frac{A(j,d-1)}{j} \ .$$

We now give a bound on the growth of A(n,d). It is obvious that A(n,d) must be monotone increasing in n, so if $j \le n$ then $A(j,d) \le A(n,d)$. We use this observation in the following derivation.

$$A(n,d) = \sum_{j=1}^{n} \frac{A(j,d-1)}{j}$$

$$\leq \sum_{j=1}^{n} \frac{A(n,d-1)}{j}$$

$$= A(n,d-1) \sum_{j=1}^{n} \frac{1}{j}$$

$$= A(n,d-1)\ H(n).$$

Iterating this recurrence on d (using $A(n,1) = 1$ as a basis) gives

$$A(n,d) \leq H(n)^{d-1},$$

where $H(j) = 1 + \frac{1}{2} + \ldots + \frac{1}{j}$. ($H(j)$ is called the j-th harmonic number; see [1].)

In the forthcoming version of this paper we tighten this bound to show the following.

Theorem 1.

$$\frac{1}{(d-1)!} H(n)^{d-1} \leq A(n,d) \leq H(n)^{d-1}.$$

Therefore, $A(n,d) = O((\ln n)^{d-1})$ for fixed d.

3. A FAST EXPECTED TIME MAXIMA ALGORITHM

So far in this paper we have considered the problem of counting the number of maxima in a set of vectors; a related problem is finding the maxima in a set of vectors. This problem has received much attention recently. Kung, Luccio and Preparata [3] give an algorithm for finding the maxima of an vectors in d-space that has worst-case running time of $O(n \ln n)$ for $d = 2$ and $O(n(\ln n)^{d-2})$ for $d \geq 3$. F. Yao [7] shows that the results in 2 and 3-space are optimal by giving a worst-case lower bound of $O(n \ln n)$ (indeed, she gives a bound that is the exact number of comparisons taken by a known algorithm for planar sets).

These results, however, deal only with the worst-case complexity of finding maxima; it is often interesting to consider also the average-case complexity. In this section we will use Theorem 1 and a general divide and conquer schema to give a fast expected time algorithm for finding maxima (this schema is investigated in detail by Bentley and Shamos [1]). The algorithm we develop here will have expected running time linear in n for vector sets drawn under the "Independent and Distinct" assumptions stated in Section 1.

Our maxima algorithm is easily described recursively. Without loss of generality, we assume that n is a power of two. To find the maxima of a set S of n vectors, divide S into two sets A and B, each containing n/2 vectors. Recursively find the maxima of A and B, calling those sets M_A and M_B, respectively. It is easy to see that the set of maximum vectors of S is the set of maxima of $M_A \cup M_B$. Therefore we can find all the maxima of S by finding the maxima of $M_A \cup M_B$; to do this we use the algorithm of Kung, Luccio and Preparata [3]. (Recursion in our original algorithm stops when n is less than some predefined constant.) The division into subproblems can be implemented on a random access computer by storing the vectors in a d by n array of scalar values. Each vector is initially represented as a pair of integers which define the left and right endpoints of a segment in the array. Division into further subsets can be accomplished by taking the arithmetic mean of the endpoints as defining two new segments, etc.; note that the division preserves randomness and can be accomplished in constant time.

The expected running time of this algorithm is easy to analyze, given that the expected number of maxima in a set of n d-vectors is $O((\ln n)^{d-1})$. Since division into subproblems can be accomplished in constant time, the recurrence describing the expected running time of our algorithm on n d-vectors is

$$T(n,d) = 2T(n/2,d) + F(n,d) \tag{3.1}$$

where F(n,d) is the expected running time of the marriage step (finding the maxima of $M_A \cup M_B$). Let i be the number of vectors in $M_A \cup M_B$. Then the running time of the marriage step using the algorithm of Kung, Luccio and Preparata is bounded above by $O(i(\ln i)^{d-2})$ for $d \geq 3$. This gives

$$F(n,d) \leq \sum_{i=1}^{n} p(i) \cdot i(\ln i)^{d-2}$$

where p(i) is the probability of there being exactly i maxima in $M_A \cup M_B$. By the fact that the number of maxima in A is independent of that in B, the expected value of i satisfies:

$$\begin{aligned} E(i) &= \sum_{i=1}^{n} p(i) \cdot i \\ &= 2\cdot(\text{expected number of maxima in a set of } n/2 \text{ d-vectors}) \\ &= 2 \cdot O((\ln \tfrac{n}{2})^{d-1}) \\ &= O((\ln n)^{d-1}). \end{aligned}$$

Therefore, by $\ln i \le \ln n$, we have

$$F(n,d) \le (\ln n)^{d-2} \sum_{i=1}^{n} p(i) \cdot i \tag{3.2}$$

$$= O((\ln n)^{2d-3}).$$

Substituting (3.2) into (3.1) gives for the running time of our algorithm the recurrence

$$T(n,d) \le 2T(n/2,d) + O((\ln n)^{2d-3}).$$

For fixed d, this recurrence is well known to have the solution

$$T(n,d) = O(n).$$

In addition to having a very fast expected running time, our algorithm also has quite a respectable worst-case performance. Note that $F(n,d)$ is always bounded above by $O(n(\ln n)^{d-2})$ for $d \ge 3$, so the worst-case running time of our algorithm is given by

$$T(n,d) = 2T(n/2,d) + O(n(\ln n)^{d-2})$$

which has the solution

$$T(n,d) = O(n(\ln n)^{d-1}).$$

Thus in the worst-case our algorithm is only a factor of ln n slower than the best known worst-case algorithm.

We summarize the main result of this section in the following theorem.

Theorem 2.

The maxima of a set of n d-vectors drawn from a distribution satisfying the "Independent and Distinct" property can be found in expected time linear in n.

4. RELATION TO CONVEX HULLS

The maximal elements of a set of vectors are a crude representation of the boundary of the set; the boundary can be more precisely defined as the boundary of the convex hull of the set. While working with the convex hull we will view the vectors as points in d-space. The convex hull of the n points is then defined as the smallest convex set containing the n points. One can get an intuitive picture of the convex hull of a planar point set by imagining the n points as n nails in a large board, with about an inch of each nail remaining above the board. The convex hull of this set can be found by taking a large rubber band, stretching it infinitely far out in all directions, then letting go. It will come to rest about certain of the nails, and the region within the rubber band is the convex hull of the set.

Given a set of n points sampled independently from some underlying probability distribution function in d-space, what is the expected number of points on the resulting convex hull? (Here we use the abbreviation "on the convex hull" to mean "on the boundary of the convex hull".) The answer to this question is of course dependent of n, d, and the underlying distribution. Santalo [4] describes a number of results for different distributions; many of these results and their original references are given in Bentley and Shamos [1]. In this section we will give an upper bound on the number of hull points for distributions satisfying our requirement of independence among the d variables. To arrive at this bound we will first show that every convex hull point is a maximum under at least one of the 2^d possible different assignments of + and - signs to the d components, and then use this fact and Theorem 1 to bound the expected number of hull points.

To show that every convex hull point is a maximum under at least one of the assignments of + and - signs, assume that there is some hull point h which is not. This implies that there is at least one point in each of h's 2^d orthants; choose one point from each orthant and call this collection P. Because values are distinct, all points in P are properly contained in their orthants. Consider now the convex hull of P; it must properly contain h. (If it contained all the points of P and not h, then it would not be a convex set.) Since h is properly contained in the convex hull of P it must also be properly contained in the convex hull of the original set. This contradicts our assumption and establishes the desired fact.

We have shown that every hull point is a maximum under at least one of the 2^d possible assignments of + and - signs to the d variables. Consider now the set of all points that are maximal under at least one of the sign assignments; call this set M. Since the expected size of M is bounded above by $2^d \cdot O((\ln n)^{d-1})$, and M contains all convex hull points, the expected number of convex hull points is certainly bounded above by that expression. Thus we have the following theorem.

Theorem 3.

The expected number of convex hull points in a point set of n points in d dimensions satisfying the "Independent and Distinct" property is bounded above by $O((\ln n)^{d-1})$.

ACKNOWLEDGMENTS

The authors gratefully acknowledge helpful discussions with Professor Michael Shamos.

BIBLIOGRAPHY

[1] Bentley, J. L. and M. I. Shamos, Divide and Conquer for Linear Expected Time, Carnegie-Mellon University Computer Science Department Report, 1977, 8 pp.

[2] Knuth, D. E., The Art of Computer Programming, Vol. 1: Fundamental Algorithms, Addison-Wesley, Reading, MA, 1973.

[3] Kung, H. T., F. Luccio and F. P. Preparata, "On Finding the Maxima of a Set of Vectors," Journal of the ACM, vol. 22, no. 4, October 1975, pp. 469-476.

[4] Santalo, L. A., "Integral Geometry and Geometric Probability," Encyclopedia of Mathematics and Its Applications, vol. 1, Addision-Wesley, 1976, 404 pp.

[5] Schkolnick, M., "A Clustering Algorithm for Hierarchical Structures," ACM Transactions on Database Systems, vol. 2, no. 1, March 1977, pp. 27-44.

[6] Schkolnick, M. and C. D. Thompson, "A Clustering Strategy for Relational Data Bases," 1977, to appear.

[7] Yao, F. F., On Finding the Maximal Elements in a Set of Plane Vectors, University of Illinois Computer Science Department Report UIUCDCS-R-74-667, July 1974, 6 pp.

INFORMATION-SINGULARITY AND RECOVERABILITY OF RANDOM PROCESSES

BRUCE HAJEK
University of California, Berkeley, California

ABSTRACT

Many new results on information-singular random processes are presented. Emphasis is given to the role of information-singular processes in the presence of other processes-- in particular--a measure of mutual information between general state space stationary random processes is used. Also, some important properties of a class of wide sense stationary random processes are deduced using novel spectral considerations.

I. INTRODUCTION

The study of information-singular random processes, or those processes which generate information slower than linearly with time, was initiated by Berger [1] in 1975. Although information-singular processes are interesting in themselves, our primary reason for studying them is to contribute to the understanding of the information generation mechanisms of all processes--not only information-singular ones. For example, the study of information-singular processes promises new theoretical tools for dealing with certain non-stationary processes. We shall consider a class of wide sense stationary processes in Section II. Information rates and other properties of these processes may be deduced from the distribution of their spectral representation processes.

Berger began studying processes which are singular even in the presence of other processes with his definition of "strong information-singularity." We have clearified his work by showing that information-singularity plus what we label "recoverability" is equivalent to strong information-singularity. In Section III we note that for stationary random processes, recoverability is a special case of subordination in Pinsker's [10] sense. In the last three sections of this paper we investigate a measure of information between stationary random processes. The measure of information we use was one of several definitions of process mutual information given by Pinsker [10] , and recently Gray [7] has established several properties, including usefulness, of this definition. We hence label it the Pinsker-Gray measure of mutual information. In Section VI we consider the implications of information-singularity and recoverability for the Pinsker-Gray measure of mutual information. For example, two stationary processes differing by an information-singular process have the same mutual information with any third jointly stationary process. A brief section on quantization is given, for quantization is important in dealing with process in arbitrary state spaces. Our study of mutual information yields a better understanding of informational properties of general processes.

This paper reports many results of [8] , ,which contains the proofs omitted here. In the remainder of this section we give some definitions and important preliminary results.

Given a set A, let A^n be the space of n-tuples from A and let A be the space of all $x=(\dots,x_{-1},x_0,x_1,\dots)$ $x_i \epsilon A$. If $(A,\underline{A})$ is a measure space, a distortion function ρ on $A \times A$ is a nonnegative measurable function on $A \times A$ such that $\rho(a,a)=0$ for all $a \in A$. Corresponding to the distortion function ρ, let ρ_m be the distortion function on $A^m \times A^m$ defined by $\rho_m(\underline{a},\underline{b})=\frac{1}{m}\sum \rho(a_i,b_i)$ for $\underline{a},\underline{b} \in A^m$. We shall often restrict our attention to complete, separable metric spaces (A,d) with Borel sigma fields and distortion functions of the form d^α where α is some fixed positive constant. Furthermore, we shall frequently assume that random processes $X=\{X_k\}$ with state space (A,d) satisfy the following condition:

Integrability Condition (IC) There exists an $a \in A$ such that, for all k, $Ed^\alpha(X_k,a)<\infty$.

A code $C=(m,e_m,d_m)$ for a measure space $(A,\underline{A})$ consists of a positive integer m, a measurable mapping $e_m: A^m \rightarrow D$, where D is some finite set, and a mapping $d_m: D \rightarrow A^m$. If M is the cardinality of D, C is a fixed-rate code with rate $\ln M/m$ nats. If $D \subset \{0,1\}^*$, where $\{0,1\}^*$ is the set of all finite length sequences of zeros and ones, then C is also a variable-rate code. Let $\ell(\underline{s})$ denote the length of $\underline{s} \in \{0,1\}^*$. If $C=(m,e_m,d_m)$ is a variable rate code and if $\underline{X}^m=(X_1,\dots X_m)$ is a random vector with values in A^m, then the average rate of C when applied to $\underline{X}^m$ is defined to be $E\ell(e_m(\underline{X}^m))$. If ρ is a distortion function on $A \times A$ the average per letter distortion of C when applied to $\underline{X}^m$ is defined to be $E\rho_m(\underline{X}^m,d_m(e_m(\underline{X}^m)))$.

Given a random process $Z= Z_k$ we will denote the random vector $(Z_p,Z_{p+1},\dots Z_m)$ by $\underline{Z}_p^m$ for integers $p \leq m$.

Let $X=\{X_k\}$ be a random process with measurable state space $(A,\underline{A})$ and let ρ be a distortion function on $A \times A$. The process X is information-singular if there exists a sequence of codes $C_n=(m=2n+1,e_n,d_n)$ such that, when C_n is applied to $\underline{X}_{-n}^n$, the rate (or average rate) and the average per letter distortion both converge to zero as $n \rightarrow \infty$. Definitions of fixed-rate information-singularity and variable-rate information-singularity result when the codes C_n are required to be fixed-rate or variable-rate codes respectively.

Fixed-rate information-singularity implies variable-rate information-singularity. We also have the following result.

Proposition 1: If X is stationary with a complete, separable metric space (A,d) as state space, if $\rho=d^\alpha$, and if the IC is satisfied, then the following are equivalent:

a) X is fixed-rate information-singular.
b) X is variable-rate information-singular.
c) X has zero entropy.
d) All the ergodic subsources of X have zero entropy, and are information singular.

The equivalence of a) and b) is actually a somewhat deep result. Proposition 1 is proved in [8] using a recent result in universal source coding (see [9]). As Berger noted, the equivalence of a) and b) when X is ergodic is easy to demonstrate.

As an elementary example, Berger showed that the complex-valued process $Z=\{Z_k=\exp(ik\Lambda)\}$, where Λ is uniform on $[-\pi,\pi)$, is fixed-rate information-singular when squared Euclidean distance is used for the distortion function. This result is immediate from the equivalence of a) and d) above, for the ergodic subsources of Z are all deterministic and hence have zero entropy.

We will next define strong information-singularity, and a related notion, which we call recoverability. We will want to consider sums and differences of random variables, so it is natural to work in a separable Banach space $(B,\| \|)$. If d is the metric on B defined by $d(a,b)=\|a-b\|$, we will also denote the Banach space by (B,d). Let $Y=\{Y_n\}$ and $N=\{N_n\}$ be a pair of possibly dependent random processes. Suppose they have (B,d) as state space and let ρ be a distortion function on $B \times B$. Then Y is strongly information-singular with respect to N if there exists a sequence of codes $C_n=(m=2n+1, e_m, d_m)$ such that when applied to $\underline{Y}_{-n}^{n} + \underline{N}_{-n}^{n}$ the rate (or average rate) of the codes tends to zero and $E\rho_m(\underline{Y}_{-n}^{n}, \hat{\underline{Y}}_{-n}^{n}) \to 0$, where $\hat{\underline{Y}}_{-n}^{n} = d_m(e_m(\underline{Y}_{-n}^{n} + \underline{N}_{-n}^{n}))$. If the codes are specified to be fixed-rate or variable-rate, we obtain the definitions of fixed-rate and variable rate information-singularity respectively.

One of the implications of strong information-singularity of Y with respect to N is that for the purpose of transmitting blocks from the "source" process $\{Y_k+N_k\}$ with small average per letter distortion, the component N is negligible. From another point of view, we could be given a "signal" process Y and a "noise" process N. The strong information-singularity of Y with respect to N implies that the process Y is deterministic in the following sense. Even when a noisy version of it is observed (the noise N may be measurement inaccuracy, for example), namely $\{Y_k+N_k\}$, the process Y can be estimated with arbitrarily small per letter distortion and then can be conveyed over any channel of positive capacity.

It is shown in [8] that strong information-singularity is equivalent to the combination of two quite unrelated concepts--information-singularity and recoverability. We will define recoverability next.

Let $Y=\{Y_k\}$ and $N=\{N_k\}$ be processes with state space (B,d), a separable Banach space, and let ρ be a distortion function on $B \times B$. The process Y is recoverable with respect to N if there exists a sequence of Borel measurable mappings $g_n: B^m \to B^m$, $m=2n+1$, such that

$$\lim_{m\to\infty} E\rho_m(\underline{Y}_{-n}^{n}, g_n(\underline{Y}_{-n}^{n} + \underline{N}_{-n}^{n}))=0.$$

If Y is a "signal" process and N is "noise", recoverability of Y with respect to Y+N implies that if one considers long enough blocks, one can estimate blocks of the signal process from a noisy version of the signal process with high accuracy.

It is easy to show that if the ditortion measure ρ is a function of the metric d then recoverability of one process with respect to another is a symmetric property.

The following proposition shows that recoverability is the link between information-singularity and recoverability. It is proved in [8].

Proposition 2: Suppose that X= X_k and N= N_k are possibly dependent random processes on a separable Banach state space (B,d) and that distortion function d , where 0. is used. Suppose also that Z satisfies the IC. Then Z is fixed-rate (variable-rate) information-singular if and only if Z is fixed-rate (variable-rate) information-singular and Z is recoverable with respect to N.

II. A CLASS OF WIDE SENSE STATIONARY PROCESSES

Suppose $\{X_k\}$ is a complex valued, wide sense stationary random process. Then, according to the spectral representation theorem, there exists a complex, orthogonal increment process $\{\beta(\lambda), -1/2 \leq \lambda \leq 1/2\}$ such that, using quadratic mean integrals,

$$X_k = \int_{-1/2}^{1/2} \exp(2\pi ik\lambda)\, d\beta(\lambda) \quad \text{a.e.}$$

The power spectral measure of $\{X_k\}$ is given by $S(d\lambda) = E(|\beta(d\lambda)|^2)$ which is also the infinitesimal covariance of the process $\{\beta(\lambda)\}$ Linear prediction theory uses only the second order properties of $\{X_k\}$. Equivalently, since the autocorrelation function of $\{X_k\}$ is determined by $S(d\lambda)$ (formally by Fourier transform) only the power spectral measure $S(d\lambda)$ is needed to apply linear prediction theory. However, as we shall see, the information rate of $\{X_k\}$ is not determined by $S(d\lambda)$, but depends on the detailed behavior of the sample paths of $\{\beta(\lambda)\}$.

Suppose now that $\{\beta(\lambda)\}$ is continuous in probability and, without loss of generality, right continuous with finite limits from the left. Then $\{\beta(\lambda)\}$ may be decomposed into the sum of two independent processes $\{\xi(\lambda)\}$ and $\{\eta(\lambda)\}$ such that $\{\xi(\lambda)\}$ is a pure jump process with independent increments and $\{\eta(\lambda)\}$ is a complex Wiener process (a continuous, independent increment process). The process $\{\xi(\lambda)\} = \{\beta(\lambda) - \eta(\lambda)\}$ is continuous in probability, right continuous and has finite left limits. This decomposition induces an additive decomposition of the process X_k, for if we define

$$Z_k = \int_{-1/2}^{1/2} \exp(2\pi ik\lambda)\, d\xi(\lambda) \quad \text{a.e.}$$

and

$$N_k = \int_{-1/2}^{1/2} \exp(2\pi ik\lambda)\, d\eta(\lambda) \quad \text{a.e.}$$

then, with probability one, $X_k = Z_k + N_k$ for all k. We will require that the process N_k have bounded spectral density. That is, if $S_N(d\lambda)$ is the power spectral measure of $\{N_k\}$, we require that $S_N([-1/2,\lambda])$ have a bounded derivative for $|\lambda| \leq 1/2$. This requirement is satisfied if, for example, $S_N(d\lambda) = \sigma_N^2\, d\lambda$ so that N_k is Gaussian white noise. Our main result regarding the processes Z and N is the following proposition.

Proposition 3: $Z = Z_k$ is strongly variable rate information-singular with respect to $N = N_k$ when squared distance is used for the distortion function. Equivalently, Z is variable-rate information-singular and Z is recoverable with respect to N.

The second assertion in Proposition 3 follows from Proposition 1. The variable -rate information-singularity of Z was demonstrated by Berger under the additional assumption that the increments of the process $\{\xi(\lambda)\}$ were stationarily distributed. The proof of Proposition 3 is given in [8,Chap. IV]. We will endeavor to give a brief plausibility argument here.

By taking discrete Fourier transforms, which preserve sums of squared distances, one can transform the problem of estimating or encoding blocks from the process Z into the problem of estimating or encoding the discrete transforms of blocks of the Z process. If large enough blocks of the Z process are considersd, the discrete transforms of the blocks tend to approximate the increments of the continuous parameter process $\{\xi(\lambda)\}$. Since $\{\xi(\lambda)\}$ is a pure jump process, its increments $\xi(\lambda+\varepsilon)-\xi(\lambda)$ tend to be very small most of the time and large on the rare occasion that a large jump of $\xi(\lambda)$ occurs in the interval $(\lambda,\lambda+\varepsilon]$. Likewise the components of the discrete transforms of blocks of Z are small with high probability. This accounts for the fact that they can be encoded with a small amount of information and with high accuracy be recovered. That is one way to demonstrate the information-singularity of Z.

Now consider recovering $\underline{Z}_{-n}^{n}$ from $\underline{N}_{-n}^{n}+\underline{Z}_{-n}^{n}$. We can again switch to the transform domain. The variables in the discrete Fourier transform of $\underline{N}_{-n}^{n}$ are gaussian and are hence, with high probability, about the same size. Since the variables of the discrete Fourier transform of $\underline{Z}_{-n}^{n}$ are either much larger or much smaller (and then, near zero) than the corresponding variables of the transform of $\underline{N}_{-n}^{n}$, the transform of $\underline{Z}_{-n}^{n}$ may be accurately recovered from the sum of the transforms (= the transform of the sum) of $\underline{Z}_{-n}^{n}$ and $\underline{N}_{-n}^{n}$. The recoverability of Z with respect to N is implicated.

It is pointed out in [8] that the regularity assumptions on N and Z may be considerably relaxed and Proposition 3 is still true.

The proof of recoverability of Z with respect to N given in [8] involves the construction of a nonlinear estimator which is based on the above heuristics. The procedure which is used is not strongly dependent on the detailed distribution of either Z or N, so Z is in some sense universally recoverable from a class of Gaussian processes.

III. SUBORDINATE RANDOM PROCESSES AND RECOVERABILITY

Suppose X and Y are variables with values in some arbitrary measure spaces $(F,\underline{F})$ and $(G,\underline{G})$. Pinsker defines [10, p. 5] X to be subordinate to Y if for all $E\in\underline{F}\times\underline{G}$ there is a $D\in\underline{G}$ so that

$$P(\{(X,Y)\in C\}\,\Delta\,\{Y\in D\})=0.$$

Intuitively, if X is subordinate to Y, the knowledge of Y is just as good as knowledge of both X and Y. In fact, the following is true. (See [8,Proposition 5.1].)

Proposition 4: Suppose X is distributed on a complete separable metric space (A,d) and that Y is distributed on a measure space $(G,\underline{G})$. Then X is subordinate to Y if and only if there is a measurable function $f:G\rightarrow A$ such that $P(X=f(Y))=1$.

A random process $X=\{X_k\}$ is defined to be subordinate to a random process $Y=\{Y_k\}$ if when viewed as a random variable X is subordinate to Y.

When the random process involved are jointly stationary, recoverability is a special case of subordination, as shown in what follows.

Proposition 5: Let $X=\{X_k\}$ and $N=\{N_k\}$ be jointly stationary with common state space (B,d), a separable Banach space. Suppose $0<\alpha$ and that $Ed^{\alpha}(X_0,a)<\infty$ for some $a\in B$. Then X is recoverable with respect to N if and only if X is subordinate to X+N.

Proposition 5 is an immediate consequence of the following result, which is proven in [8].

Proposition 6: Let $X=\{X_k\}$ and $Y=\{Y_k\}$ be jointly stationary random processes with respective state spaces (A,d), a complete separable metric space, and $(S,\underline{S})$, an arbitrary measure space. Suppose X satisfies the IC for some $\alpha>0$. Then X is subordinate to Y if and only if there is a sequence of measurable functions $f_n : S^n\to A^n$ such that, with $\rho=d^{\alpha}$,

$$E\rho_n(\underline{X}_1^n,\ f_n(\underline{Y}_1^n)) \to 0 \text{ as } n\to\infty.$$

IV. QUANTIZATIONS OF PROCESSES

Let $(S,\underline{S})$ be an arbitrary measure space. Denote by $\underline{S}^{\infty}$ the smallest σ-field of subsets of S^{∞} which contains the cylinder sets, i.e., sets of the form

$$\{\underline{x}\in S^{\infty} : (x_{m+1},\dots x_n)\in B\} \quad B\in\underline{S}^{m-n}.$$

Let $T:S^{\infty}\to S^{\infty}$ be the unit shift to the left on S^{∞}. T is measurable using the σ-field $\underline{S}^{\infty}$ on S^{∞}.

Suppose Z is a random variable with values in the measure space $(S,\underline{S})$. A *quantizer* on $(S,\underline{S})$ is any measurable function $\gamma:S\to\hat{S}$ where $\hat{S}$ is some finite set, and the $\hat{S}$ valued random variable $\gamma(Z)$ is called a *quantization of Z* (corresponding to the quantizer γ). If $X= X_k$ is a random process with state space $(S,\underline{S})$, then X may also be viewed as a random variable with values in the measurable space $(S^{\infty},\underline{S}^{\infty})$. If γ is a quantizer on $(S^{\infty},\underline{S}^{\infty})$, then the random process $\hat{X}=\{X_k=\gamma\circ T^k(X)\}$ is called a *process quantization* of X. If $\gamma(X)$ is subordinate to X_0, then γ is a *one-dimensional quantization* of X.

If X is a stationary random process then process quantizations of X are also stationary, and they inherit information-singularity:

Proposition 7: If X is a stationary random process with a complete, separable metric state space and if X is information-singular under distortion d^{α}, then all process quantizations of X are also information-singular.

This result is first proven in [8] and it easily implies that such information-singular random processes have zero entropy. The result is intuitively pleasing though tricky--the difficulty is that if a sequence of random variables converges to a random variable, quantizations of the variables need not converge. In fact, the preservation of information-singularity even under one-dimensional quantization fails for some real-valued wide sense stationary random processes.

V. THE PINSKER-GRAY MEASURE OF PROCESS MUTUAL INFORMATION

If Y and Z are random variables on a finite set then the entropy of Y is defined by

$$H_o(Y) = \sum_y -P(Y=y)\log P(Y=y)$$

and the mutual information between Y and Z is

$$I_o(Y;Z) = H_o(Y) + H_o(Z) - H_o(Y;Z).$$

If $\xi = \{\xi_k\}$ is a stationary random process such that ξ_k takes on finitely many values for each k then the entropy of the process ξ is defined by

$$H_o(\xi) = \lim_{n\to\infty} \frac{1}{n} H_o(\xi_1, \ldots \xi_n).$$

The limit always exists and is finite. If $\xi = \{\xi_k\}$ and $\eta = \{\eta_k\}$ are jointly stationary and have finite state spaces then the process mutual information between ξ and η is

$$I^{(\infty)}(\xi;\eta) = H_o(\xi) + H_o(\eta) - H_o(\xi,\eta).$$

To define mutual information between continuously distributed random variables and processes (which often have infinite entropy), quantizations are used. If Y and Z is an arbitrary pair of random variables, the mutual information between them may be defined as

$$I_o(Y;Z) = \sup I_o(\hat{Y};\hat{Z})$$

where the supremum is over all quantizations of Y and Z and I_o was already defined for discrete random variables. If $\xi = \{\xi_k\}$ and $\eta = \{\eta_k\}$ are jointly stationary random processes then the Pinsker-Gray definition of mutual information between the processes is

$$I(\xi;\eta) = \sup I^{(\infty)}(\hat{\xi};\hat{\eta}) = \sup \lim_{n\to\infty} \frac{1}{n} I(\hat{\xi}_1^n; \hat{\eta}_1^n)$$

where the supremum is over all process quantizations of ξ and η. It is essential here that the supremum over quantizations is taken after the limit as $n\to\infty$. We believe that this order yields the most useful measure of information between random processes. We will list some properties of this definition, most of which other measures of mutual information lack.

Proposition 8:

1. If y and Z are jointly stationary random processes then $I(Y;Z) \le \infty$ is always defined.

2. Virtually all the formulas for mutual information in the discrete random variable case carry over. For example, if Y,Z,N, and W are jointly stationary random processes then we have $I(Z;Z) = H(Z)$ (where $H(Z)$ is the standard definition of process entropy), $I(Z;N) \le \min(H(Z), H(N))$, and a generalization of Dobrushin's formula [10, p. 45]:

$$I((Y,Z);N) + I(Y;Z) = I(Y;(N,Z)) + I(N;Z).$$

If (Y,Z) and (W,N) are independent, then

$$I(Y,W;Z,N) = I(Y;Z) + I(W;N).$$

3. If X,Y, and Z are jointly stationary random processes and if X is subordinate to Y then

$$I(X;Z) \leq I(X,Y;Z) = I(Y;Z).$$

4. Gray has shown [7,Theorem 4.7] that the mutual information I between jointly stationary random processes is the average of the mutual informations of the ergodic modes. (i.e., I admits an ergodic decomposition.)

5. I has been used to provide the simplest known proof of the basic coding theorem for ergodic sources (see [7]), demonstrating the operational significance of I.

6. Many seemingly different methods of defining I lead to the same definition (see [8]).

VI. IMPLICATIONS OF INFORMATION-SINGULARITY AND RECOVERABILITY FOR MUTUAL INFORMATION

The following proposition deals with information-singularity.

Proposition 9: Suppose that Z^{I}, Z^{II}, N, and Y are jointly stationary random processes, and that Y is distributed on an arbitrary state space.

a) If Z^{I} and Z^{II} are distributed on a separable Banach space (B,d), if they satisfy the IC, and if the process $N = Z^{I} - Z^{II} = \{Z_k^{I} - Z_k^{II}\}$ is information-singular, then $I(Z^{I};Y) = I(Z^{II};Y)$.

b) More generally, if Z^{I}, Z^{II}, N, and Y are all distributed on arbitrary state spaces, if N is a zero entropy process, and if the pair processes (Z^{I},N) and (Z^{II},N) are mutually subordinate, then $I(Z^{I};Y) = I(Z^{II};Y)$.

Proof: Under the conditions of a), N satisfies the IC and hence has zero entropy (Proposition 1). So a) follows from b) which is proved next.

Under the conditions of b), we obtain by Proposition 8.2 and 8.3 that

$$\begin{aligned} I(Z^{I};Y) \leq I(Z^{II},N;Y) &= I(Z^{II},N;Y) + I(Z^{II};N) \\ &= I(N;Z^{II},Y) + I(Z^{II};Y) \\ &= I(Z^{II};Y). \end{aligned}$$

The inequality $I(Z^{II};Y) \leq I(Z^{I};Y)$ follows likewise, completing the proof of Proposition 9.

The following proposition deals with recoverability.

Proposition 10: Suppose $Y = \{Y_k\}$, $Z = \{Z_k\}$, and $N = \{N_k\}$ are jointly stationary random processes and that Y has an arbitrary measurable state space, while Z and N are each distributed on a separable Banach space. Assume also that Z satisfies the IC for some $\alpha > 0$. Then, if Z (or equivalently N) is recoverable from N+Z, then

a) $I(Z+N,Y) = I(Z,N\,;Y)$

b) $I(Z+N;Y) = I(Z;Y)$ if N is independent of (Y,Z)

c) $I(Z+N;Z) = H(Z) \leq \infty$.

REFERENCES

1. T. Berger, "Information Singular Process," IEEE Trans. on Information Theory, Vol. IT-20, pp. 502-511, Sept. 1975

2. T. Berger, Rate Distortion Theory, Englewood Cliffs, N. J., Prentice-Hall, 1971.

3. H. Cramer and M. R. Leadbetter, Stationary and Related Stochastic Processes, New York, Wiley, 1967.

4. R. G. Gallager, Information Theory and Reliable Communication, New York, Wiley, 1968.

5. I. I. Gihman and A. V. Skorohod, The Theory of Stochastic Processes, New York, Springer-Verlag, 1974.

6. I. I. Gihman and A. V. Skorohod, The Theory of Stochastic Processes II, New York, Springer-Verlag, 1975.

7. R. M. Gray, "Mutual Information Rate, Distortion, and Quantization in Metric Spsces," Submitted to IEEE Trans. on Information Theory, April 1977.

8. B. E. Hajek, "Information Singularity and Recoverability of Random Processes," M. S. Thesis, University of Illinois, 1977. To become a technical report.

9. D. L. Neuhoff, R. M. Gray and L. D. Davisson, "Fixed Rate Universal Block Source Coding with a Fidelity Criterion," IEEE Trans. on Information Theory, Vol. IT-20, pp. 511-523, Sept. 1975.

10. M. S. Pinsker, Information and Information Stability of Random Variables, Moscow, Akad. Nauk., 1960. (English translation by A. Feinstein, Holden Day, San Francisco, 1964.)

ACKNOWLEDGEMENTS

I owe thanks to Prof. M. B. Pursley and Dr. Ken Mackenthun for helpful suggestions. This work was supported by the National Science Foundation under Grant ENG75-20864 and through a graduate research fellowship.

LINEAR DIMENSIONALITY REDUCTION OF STATIONARY VECTOR GAUSSIAN PROCESSES

P. Papantoni-Kazakos*
Rice University, Houston, Texas

R. Li*
IBM, Poughkeepie, New York

D. Kazakos**
SUNY at Buffalo, Amherst, New York

ABSTRACT

Expressions for the Chernoff distance and divergence between two stationary Gaussian processes are derived, when the number of sample points is large. The expressions contain only the two spectral densities. We are then posing the problem of finding the best linear transformation that maximizes the transformed divergence and Chernoff distance. Such a transformation preserves discrimination optimally.

*Research supported by Air Force Grant AFOSR 77-3156.

**Research supported by National Science Foundation Grant ENG 76-20295.

I. INTRODUCTION

The subject of data reduction has been of great interest to statisticians and system scientists. Extensive literature has been devoted to the subject due to its practical significance. We will not attempt to give any account of the extensive work that has been done. Such an account can be found in [1].

The basic novelty of the present work is the availability of a very large number n of samples produced from one of two stationary Gaussian random processes. All of the existing feature selection approaches become intractable for large n, basically due to the requirement of finding eigenvalues and eigenvectors of nxn covariance matrices. We are bypassing this problem by using the spectra of the random processes. In [5], [6] the problem of dimensionality reduction using a frequency domain approach and different objectives was considered. Here, the objective is the efficient discrimination between the two processes on the basis of the reduced data and using the stated two distance measures as criteria. The above distance measures are related to the probability of error in detection between the two hypotheses through upper and lower bounds. Therefore, their maximization is most relevant for optimized classification performance. One application of the results of the present paper is in the reduction of Remotely Sensed satellite data.

Results

Let $x^n=(x_1 . . . x_n)$ be a vector consisting of samples from one of two stationary Gaussian processes with zero means, and spectral densities

$f_1(\lambda)$, $f_2(\lambda)$ under hypotheses H_1, H_2 correspondingly. Let R_{1n}, R_{2n} be the covariance matrices of x^n under H_1, H_2:

$$R_{mn} = E[(x^n)^t x^n | H_m], \quad m = 1,2 \tag{1}$$

R_{mn} has elements $\{r_m(|s-j|)\}$, and

$$f_m(\lambda) = \sum_{k=-\infty}^{+\infty} r_m(k)\exp(ik\lambda) \tag{2}$$

We assume

$$0 < u_1 < f_m(\lambda) < u_2 < +\infty, \qquad \lambda \varepsilon [0,2\pi] \tag{3}$$

We denote by C_n, D_n correspondingly the Chernoff distance and Divergence between classes H_1, H_2 and on the basis of the n-dimensional observation vector x^n. They are given by the expressions: [7], [8]

$$2C_n \overset{0<s<1}{=} \log|sR_{1n} + (1-s)R_{2n}| - s\log|R_{1n}| - (1-s)\log|R_{2n}| \tag{4}$$

$$2D_n = \text{trace}[R_{1n}^{-1} R_{2n} + R_{2n}^{-1} R_{1n} - 2I] \tag{5}$$

The following bounds of the probability of misclassification are available: [7]

$$P_e(n) < 2^{-1}\exp\{-C_n\} \tag{6a}$$

$$P_e < 2^{-1}(D_n/4)^{-1/4} \tag{6b}$$

Our objective is to seek convenient expressions for the limits:

$$C = \lim_{n\to\infty} n^{-1}C_n \quad , \quad D = \lim_{n\to\infty} n^{-1}D_n \tag{7}$$

The significance of C, D is that we can use them to express (6) in the form:

$$P_e(n) < 2^{-1}\exp\{-n(C+O(1))\} \quad , \; O(1) \to 0 \quad \text{as} \quad n \to \infty \tag{8a}$$

$$P_e(n) < 2^{-1}[n(D/4 + O(1))]^{-1/4} \tag{8b}$$

The basic tool for calculating C, D will be the following theorem: [2], [3] Theorem

Let R_n be an nxn Toeplitz matrix, with elements $\{r(|i-j|)\}$ and eigenvalues $\{\lambda_{n,k}, k=1, \ldots, n\}$. Let

$$f(\lambda) = \sum_{k=-\infty}^{+\infty} r(k)\exp(ik\lambda), \text{ and assume}$$

$$0 < u_1 < f(\lambda) < u_2 < +\infty \quad , \quad \lambda \in [0,2\pi]$$

Let F(x) be a function continuous for $x \in [u_1, u_2]$. Then

$$\lim_{n\to\infty} n^{-1} \sum_{k=1}^{n} F(\lambda_{n,k}) = (2\pi)^{-1} \int_0^{2\pi} F[f(\lambda)]d\lambda$$

The above theorem concerns the asymptotic distribution of eigenvalues of Toeplitz matrices, and it has been used heavily in Information Theory. We will now start with the calculation of C. Let $\{\lambda^m_{n,k}, k=1, \ldots, n\}$ be the eigenvalues of R_{mn}, m = 1,2. Then,

$$n^{-1}\log|R_{mn}| = n^{-1} \sum_{k=1}^{n} \log\lambda^m_{n,k}$$

We apply the theorem with F(x) = log x; and the result is:

$$\lim_{n\to\infty} n^{-1}\log|R_{m,n}| = \lim_{n\to\infty} n^{-1} \sum_{k=1}^{n} \log\lambda^m_{n,k} = (2\pi)^{-1} \int_0^{2\pi} \log f_m(\lambda)d\lambda \tag{9}$$

Similarly, let $\{q_{n,k}, k=1, \ldots, n\}$ be the eigenvalues of the matrix $sR_{1n} + (1-s)R_{2n}$. This matrix is Toeplitz with elements

$$r(|i-j|) = sr_1(|i-j|) + (1-s)r_2(|i-j|)$$

Hence, the corresponding spectral density is:

$$f(\lambda) = sf_1(\lambda) + (1-s)f_2(\lambda)$$

Using the same theorem, we have:

$$\lim_{n\to\infty} n^{-1}\log|sR_{1n}+(1-s)R_{2n}| = \lim_{n\to\infty} n^{-1} \sum_{k=1}^{n} \log q_{n,k} =$$

$$= (2\pi)^{-1} \int_0^{2\pi} \log[sf_1(\lambda) + (1-s)f_2(\lambda)]d\lambda \tag{10}$$

Collecting terms, we obtain:

$$2C = (2\pi)^{-1} \int_{o}^{2\pi} \log[sf_1(\lambda) + (1-s)f_2(\lambda)]-s \log f_1(\lambda)-(1-s)\log f_2(\lambda)\}d\lambda$$

The convexity of the log function makes the integrand non-negative. C is zero if $f_1(\lambda) = f_2(\lambda)$.

Next, we will calculate D. Different arguments are needed. The quantity to compute now is

$$\lim_{n\to\infty} n^{-1} \mathrm{trace} R_{1n} R_{2n}^{-1} = \lim_{n\to\infty} n^{-1} \sum_{i=1}^{n} q_{ni}$$

where $\{q_{ni}\}$ = eigenvalues of $R_{1n}R_{2n}^{-1}$

We say that two sequences of nxn matrices A_n, B_n are asymptotically equivalent [2], $A_n \sim B_n$, if

$$\lim_{nn\to\infty} |A_n - B_n| = 0,$$

where $|\cdot|$ is a weak norm [2]. We will use now the following theorem:

Theorem [2]

Let $m < a_{n,i}, b_{n,i} < M$, i=1, . . .,n be the eigenvalues of two Hermitian matrix sequences, A_n, B_n, bounded by [m,M]. Let F(x) be continuous on [m,M]. Assume $A_n \sim B_n$. Then

$$\lim_{n\to\infty} n^{-1} \sum_{k=1}^{n} F[a_{n,k}] = \lim_{n\to\infty} n^{-1} \sum_{k=1}^{n} F[b_{n,k}]$$

In another theorem [3], it is shown that given a Toeplitz matrix, R_n with corresponding $f(\lambda)$ such that $0 < u_1 < f(\lambda) < u_2 < +\infty$, we can find a circulant matrix. S_n, such that $R_n \sim S_n$. Hence, we find $S_{1n} \sim R_{1n}$ and $S_{2n} \sim R_{2n}$. Then, we can show after some manipulations, that

$$S_{1n}S_{2n}^{-1} \sim R_{1n}R_{2n}^{-1}$$

But the eigenvalues of $S_{1n}S_{2n}^{-1}$ are $\{f_1(2\pi kn^{-1}) \cdot f_2^{-1}(2\pi kn^{-1})\}$, and we can use the previous theorem to conclude: (taking F(x) = x)

$$\lim_{n\to\infty} n^{-1} \mathrm{trace}\, R_{1n}R_{2n}^{-1} = (2\pi)^{-1} \int_{o}^{2\pi} f_1(\lambda) f_2^{-1}(\lambda) d\lambda \tag{12}$$

We can now write the final expression for D;

$$2D = (2\pi)^{-1} \int_{o}^{2\pi} [\frac{f_1(\lambda)}{f_2(\lambda)} + \frac{f_2(\lambda)}{f_1(\lambda)} - 2] d\lambda \tag{13}$$

Using (11), (13), we can pose now a class of optimal dimensionality reduction problems. Consideration of both D and C is waranted, because of the unknown begree of tightness of the bounds, (6). Passing the process through any time invariant filters leaves (13), (11) invariant.

Let $Z_n = [z_{1n} z_{2n} \ . \ . \ . \ z_{mn}]^t$ be the nth sample of an m-dimensional stationary Gaussian random process, distributed according to one of two hypotheses, H_1, H_2. Let

$$E[z_{kn} z_{is} | H_u] = r^u_{ki}(|n-s|), \quad u = 1,2 \qquad k,i = 1, \ . \ . \ . \ , m$$

be the cross-correlations. The corresponding spectral densities are:

$$f^u_{ki}(\lambda) = \sum_{q=-\infty}^{\infty} r^u_{ki}(q)\exp(jq\lambda), \qquad u = 1,2$$

$F_u(\lambda)$ = mxm matrix with elements $\{f^u_{ki}(\lambda)\}$

The vector process $\{Z_n\}$ is reduced to a scalar one $\{x_n\}$ by the linear transformation:

$x_n = A^t Z_n$, A = feature selecting vector.

$\{x_n\}$ is a stationary Gaussian process, with spectral density:

$$f_u(\lambda) = A^t F_u(\lambda)A, \quad u = 1,2 \tag{14}$$

under hypothesis H_1, H_2 correspondingly.

The transformed asymptotic Chernoff distance C and divergence D are now easily expressed as functions of the transformation A:

$$2C = (2\pi)^{-1} \int_0^{2\pi} \log[sA^t F_1(\lambda)A + (1-s)A^t F_2(\lambda)A] - s \log A^t F_1(\lambda)A - \tag{15}$$

$$-(1-s)\log A^t F_2(\lambda)A\}d$$

$$2D = (2\pi)^{-1} \int_0^{2\pi} [\frac{A^t F_1(\lambda)A}{A^t F_2(\lambda)A} + \frac{A^t F_2(\lambda)A}{A^t F_1(\lambda)A} - 2]d\lambda \tag{16}$$

We are interested in maximizing C, D over the choice of A. Necessary conditions for extremizing C, D are:

$$\nabla_A C = 0, \quad \nabla_A D = 0 \tag{17}$$

In the maximizing of C, a fixed value of $s \in [0,1]$ has been assumed. Hence, we can further maximize C over the choice of s, and pick the value of A resulting from the double maximization.

REFERENCES

[1] L. Kanal, "Patterns in Pattern Recognition"1968-1974", IEEE Trans. on Infor. Theory, Vol. IT-20, pp 697-722, Nov. 1974.

[2] R. M. Gray, "On the Asymptotic Eigenvalue Distribution of Toeplitz Matrices", IEEE Trans. on Infor. Theory, Vol. IT-18, pp 725-730, Nov. 1972.

[3] U. Grenander and G. Szego, Toeplitz Forms and Their Applications. Berkeley, Calif.: Univ. California Press, 1958.

[4] R. M. Gray, "Information Rates of Autoregressive Processes", IEEE Trans. on Infor. Theory, Vol. IT-16, pp 412-421, July 1970.

[5] D. R. Brillinger, The Canonical Analysis of Stationary Time Series, in Multivariate Analysis II, Editor P. R. Krishnaiah, Academic Press, New York, pp 331-350, 1969.

[6] M. B. Prestley, T. S. Rao and H. Tong, "Applications of Principal Component Analysis and Factor Analysis in the Identification of Multivariable Systems", IEEE Trans. on Autom. Control, Vol. AC-19, pp 730-735, Dec. 1974.

[7] T. Kadota and L. A. Shepp, "On the best Finite Set of Linear Observables for Discriminating Two Gaussian Signals", IEEE Trans. on Infor. Theory, Vol. IT-13, pp 278-285, April 1967.

[8] T. Kailath, "The Divergence and Bhattacharyya Distance Measures in Signal Selection", IEEE Trans. Comm. Tech., Vol. COM-15, pp 52-60, Feb. 1967.

[9] T. L. Henderson and D. G. Lainiotis, "Comments on Linear Feature Extraction", IEEE Trans. on Infor. Theory, Vol. IT-15, pp 728-730, Nov. 1969.

[10] T. L. Henderson and D. G. Lainiotis, "Application of State-Variable Techniques to Optimal Feature Extraction-Multichannel Analog Data", IEEE Trans. on Infor. Theory, Vol. IT-16, pp 396-406, July, 1970.

ON REDUCING MULTIPLICATIONS IN AN EXTENDED GENERALIZED WIENER FILTER*

KUNG YAO
Department of System Science
University of California
Los Angeles, CA 90024

ABSTRACT

In this paper, we formulate a new extended generalized Wiener filter problem for signal and image processing. Several approaches are used to reduce the multiplicative computational requirements needed for their implementation. For a specific example of Markov signal in additive white noise, various new results show that there is considerable possibility of reducing the number of multiplications with only minor loss of performance.

I. INTRODUCTION

Recent advances in microelectronics have made dedicated microprocessors and special-purpose hardware devices attractive for many signal processing tasks. In contrast to general purpose computers, these signal-processing devices are generally constrained in memory, speed, and certain arithmetical operations. Typically, these devices operate only with fixed-point arithmetical capability. The execution time for performing a multiplication/division can be more than an order of magnitude greater than that for performing an addition/subtraction. Thus, given certain basic constraints in hardware capability due to the state of the art in technology and to economic cost, and given certain basic signal-processing requirements, it is clear that we need to analyze the various algorithms which may be used.

In this paper, we shall consider a class of signal filtering problems and various approaches which may be used to reduce the multiplicative computational requirements in implementing them. The problem of generalizing Wiener filtering and efficient algorithms for possible implementation is of considerable practical as well as theoretical interest ([1]-[4]). There has been much recent interest in practical image processing and compression with space and satellite applications. At the theoretical level, these filtering problems may provide some concrete motivation for the construction of models that can be used in analyzing the problem of computational complexity in signal processing.

In Section II, we define the new concept of an extended generalized Wiener filter. Besides generalizing the existing theory, another motivation for this new concept is based on its possible usefulness in studying the issue of trade-off in complexity versus performance. The resulting

---

*This work was supported by the Electronics Program of the Office of Naval research.

optimum vector and scalar filters, as well as associated mean-square errors, are presented. In Section III, various specific algorithms are given which permit the reduction of multiplicative computational requirements. Generally, as the multiplicative requirement goes down, the performance also goes down. However, for a specific practical case of interest, of a Markov signal contaminated by a white noise, various new results tend to show that there is considerable possibility of reducing multiplicative requirements with only minor loss of performance. In Section IV, a brief conclusion is drawn.

II. EXTENDED GENERALIZED WIENER FILTER

In Fig. 1, we consider an extended generalized Wiener filter. The input $N \times 1$ column vector $Z = X + W$ consists of the signal vector X plus a white noise vector W. Both X and W are assumed to have zero means and are uncorrelated. The covariance matrices of X and W are denoted by Λ_X and $\Lambda_W = kI$, respectively.

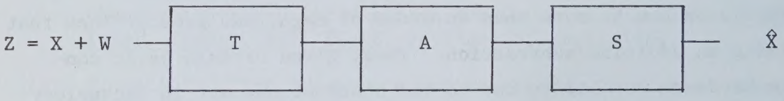


Fig. 1: An extended generalized Wiener filter

In the classical Wiener filter, T and S are taken to be identity operators and A is taken to be a linear operator such that $\hat{X} = AZ$ is close to X in the mean-square error sense. In a generalized Wiener filter ([1]-[3]), T is taken to be S^{-1}, S is required to be an orthogonal matrix, and $\hat{X} = SATZ$ is required to be close to X in the mean-square error sense. As discussed in [1]-[3], the requirements of $T = S^{-1}$ and $SS' = I$ can allow the replacement of the optimum A_o by some other sparse matrix $\tilde{A}$ (possibly diagonal) such that the multiplicative computational requirement of $S\tilde{A}T$ is considerably less than that of SA_oT.

In the extended generalized Wiener filter $B = SAT$, we allow T to be an arbitrary invertible real-valued matrix but still require S to be an arbitrary invertible orthogonal matrix. Then the mean-square error ϵ^2 is given by

$$\epsilon^2 = E\{(X-\hat{X})'(X-\hat{X})\}$$

$$\epsilon^2 = E\{F'A'AF\} - 2E\{F'A'S'X\} + \mathrm{Tr}\Lambda_X \tag{2.1}$$

where $F = TZ$. If A is required to have all non-zero elements, then by setting the gradient of ϵ^2 to zero, we obtain the optimum filter matrix A_o given by

$$A_o = S'\Lambda_X(\Lambda_X + \Lambda_W)^{-1} T^{-1} \tag{2.2}$$

and the optimum overall system matrix

$$B_o = SA_oT = \Lambda_X(\Lambda_X + \Lambda_W)^{-1} \quad . \tag{2.3}$$

The minimum mean-square error ϵ^2_{Min}, with (2.2) substituted into (2.3), is given by

$$\epsilon^2_{Min} = Tr\{\Lambda_X - B_o\Lambda_X\} \quad . \tag{2.4}$$

We note that, while T is allowed to be an arbitrary invertible real-valued matrix, the optimum overall system matrix B_o and the minimum mean-square error ϵ^2_{Min} are independent of S and T. That is, B_o and ϵ^2_{Min} given here are identical to those in the classical Wiener filter case and the generalized Wiener filter case ([3; pp.185-186]).

Since Λ_X and Λ_W in (2.3) are known, $\Lambda_X(\Lambda_X + \Lambda_W)^{-1}$ can be pre-computed. Thus, B_o is a specified $N \times N$ matrix. Then the implementation of the optimum mean-square estimate $\hat{X} = B_oZ$ normally needs N^2 multiplications. However, by using an algorithm of Winograd [5], it is possible to reduce this computation to approximately $N^2/2$ multiplications. Now, express

$$\hat{X} = \begin{bmatrix} \hat{X}_1 \\ \hat{X}_2 \\ \vdots \\ \hat{X}_N \end{bmatrix} = B_oZ = \begin{bmatrix} B^1 \\ B^2 \\ \vdots \\ B^N \end{bmatrix} Z \tag{2.5}$$

where B^i, $1 \le i \le N$, is the i^{th} $1 \times N$ row vector of B_o. Then $\hat{X}_i$ can be expressed as

$$\hat{X}_i = B^iZ = \sum_{j=1}^{N/2} (b^i_{2j-1} + z_{2j})(b^i_{2j} + z_{2j-1}) - \xi^i - \eta \ , \quad 1 \le i \le N \ , \tag{2.6}$$

where b^i_j is the j^{th} element of B^i, and

$$\xi^i = \sum_{j=1}^{N/2} b^i_{2j-1} \, b^i_{2j} \tag{2.7}$$

$$\eta = \sum_{j=1}^{N/2} z_{2j-1} \, z_{2j} \quad . \tag{2.8}$$

Thus, $\hat{X}$, when implemented in the form of (2.6), (2.7) and (2.8), needs N/2 multiplications in (2.8), no "real time" multiplication in (2.7) since ξ^i can be pre-computed and stored in memory, and N/2 multiplications in each of the summations in (2.6). Thus, the total multiplications M needed for this optimum filtering is given by

$$M = \frac{N}{2} + N \times \frac{N}{2} = \frac{N}{2}(N+1) \simeq \frac{N^2}{2} \quad . \tag{2.9}$$

A filter that uses all N^2 elements of A or B is called a full vector filter. The multiplications needed are on the order of N^2 operations. For large N , it is particularly relevant to find operations in which the multiplicative operation grows slower with N . One possible approach is to force A to be a diagonal matrix D and find T and S that are "easy" to implement in some sense.

Now, we want to find the optimum diagonal matrix D_o in the extended generalized Wiener filter sense. Let D be an N × N diagonal matrix with $d_{ij} = 0$, $i \neq j$, and d_{ii} arbitrary for $1 \leq i \leq N$. Then the gradient of the quadratic form X'D'DX is given by

$$\nabla_D \{X'D'DX\} = 2\mathscr{D}\{DXX'\} \tag{2.10}$$

where $\mathscr{D}\{C\}$ yields a diagonal N × N matrix with diagonal elements equal to the diagonal elements of C . Similarly,

$$\nabla_D \{X'DV\} = \mathscr{D}\{VX'\} \quad . \tag{2.11}$$

We note that the gradient operators $\nabla_D\{\cdot\}$, when D is a diagonal matrix, should be those given in (2.10) and (2.11) and are not the same as the usual definition of $\nabla_A\{\cdot\}$ evaluated with $a_{ij} = 0$ for $i \neq j$. In any case, start with the ϵ^2 expression in (2.1), set $\nabla_D \epsilon^2 = 0$, and use (2.10) and (2.11). Then we have

$$\mathscr{D}\{D_o(\tilde{\Lambda}_X + \tilde{\Lambda}_W)\} = \mathscr{D}\{\tilde{\Lambda}_U\} \quad , \tag{2.12}$$

$$\tilde{\Lambda}_U = S'T'^{-1} \tilde{\Lambda}_X \quad , \quad \tilde{\Lambda}_X = T\tilde{\Lambda}_X T' \quad , \quad \tilde{\Lambda}_W = T\Lambda_W T' \quad . \tag{2.13}$$

Let d_{ii} be the i^{th} diagonal element of the optimum diagonal matrix D_o ; then

$$d_{ii} = \frac{\tilde{\Lambda}_U(i,i)}{\tilde{\Lambda}_X(i,i) + \tilde{\Lambda}_W(i,i)} \quad , \quad 1 \le i \le N \ , \tag{2.14}$$

where $\tilde{\Lambda}(i,i)$ denotes the i^{th} diagonal element of $\tilde{\Lambda}$. We note that the expression in (2.14) is a generalization of Eq.(8.6.2) in ([3]; p. 191]) when T is required to be the inverse of S. The resulting mean-square error associated with D_o of Eq.(2.14) is given by

$$\epsilon^2 = \mathrm{Tr}\{\Lambda_X - SD_oT\Lambda_X\} \quad . \tag{2.15}$$

III. TECHNIQUES FOR REDUCING MULTIPLICATIVE COMPUTATIONAL COMPLEXITY

In this section, we shall consider some previously known as well as some new techniques for reducing the multiplicative computational complexity requirements of the extended generalized Wiener filter. From Fig. 1, if A is restricted to be a diagonal matrix D, then there are at most N multiplications in the operation involving D. If T and S are not chosen appropriately, then there may be another N^2 multiplications for each of the T and S operations. Thus, clearly it is crucial to select T and S carefully. Furthermore, unlike the case of the optimum A_o matrix in (2.2) with ϵ^2_{Min} given in (2.4), the present ϵ^2 given in (2.14) clearly depends on T and S. Thus, we want not only to find T and S that are easy to compute in the sense of needing fewer multiplications, we also want these T and S to have the ability to make the input to D as uncorrelated as possible in order to obtain a small ϵ^2.

It is known that, if T is taken to be constructed from the eigenvectors of B_o given by (2.3), and S is the inverse of T, then D_o given by (2.14) realizes the minimum mean-square error. In general, these eigenvectors are dependent on Λ_X and Λ_W and no "fast" algorithms exist (see [4]). Recently, for Markov signals, some forms of "fast" algorithms have been found (see [4]). However, in this paper, we shall not deal with these algorithms.

It is well known that various discrete-time transforms have "fast" algorithms. In an FFT, there are approximately $N \log_2 N$ complex multiplications; in a DCT, there are approximately $2N \log_2 N$ complex multiplications; in a Haar transform, there are N multiplications; and in a Walsh-Hardarmard transform, there are 0 multiplications. Thus, if S (or T) uses any of these (or other) transforms, the total number of multiplications M in the SD_oT operations can be easily evaluated.

In order to make these discussions meaningful, it is necessary to consider the associated mean-square error. Thus, we shall consider a specific example of $N=8$ for a Markov signal with correlation coefficient $\rho = .9$ and SNR of $k=1$. This example has been used by others (see [2]-[3]) and can provide a comparison of relative efficiencies in multiplicative computational complexity versus mean-square errors.

In Table 1, the first column gives the conditions on the A, T, and S matrices. The second column yields the associated mean-square error, and the third column yields the number of multiplications needed to perform the required filtering operation. The symbol $(\ )^*$ denotes the number of complex multiplications. In all four case 1 situations, $T=S=I$.

TABLE 1

Mean-square errors versus number of multiplications
for various conditions on A, T, and S matrices
for a Markov signal of $\rho = .9$, SNR = 1, and $N=8$.

	Conditions on A, T and S	MSE(ϵ^2)	M = number of Multiplications
1a.	$A = A_o$; $T=S=I$ (regular implementation)	2.0263	N^2
1b.	$A = A_o$; $T=S=I$ ((2.5)-2.8))	2.0263	$N^2/2$
1c.	$A = \mathscr{D}\{A_o\}$; $T=S=I$	4.	N
1d.	$A = \{\mathscr{D}A_o\}_{(1)}$; $T=S=I$	4.	0
2a.	$A = D_o$; $T=S'=$DCT	2.03667	$N+(4N\log_2 N)^*$
2b.	$A = D_o$; $T=$QDCT; $S'=$DCT	2.10348	$N+(2N\log_2 N)^*$
2c.	$A = \{D_o\}_{(1,1/2,1/4,1/8,1/16)}$; $T=S'=$DCT	2.05106	$0+(4N\log_2 N)^*$
2d.	$A = \{D_o\}_{(1,1/2,1/4)}$; $T=$QDCT ; $S'=$DCT	3.80662	$0+(2N\log_2 N)^*$
3a.	$A = D_o$; $T=S'=$WHT	2.11885	N
3b.	$A = \{D_o\}_{(1,1/2,1/4,1/8)}$; $T=S'=$WHT	2.12308	0
4a.	$A = D_o$; $T=S'=$HT	2.11983	N + 2N
4b.	$A = D_o$; $T=$QHT; $S'=$HT	2.16230	N + N
4c.	$A = \{D_o\}_{(1,1/2,1/4,1/8)}$; $T=$ HT; $S'=$HT	2.12308	0 + 2N
4d.	$A = D_{o(1,1/2,1/4,1/8)}$; $T=$QHT; $S'=$HT	2.17004	0 + N

Case 1a represents a regular implementation of an $N \times N$ matrix multiplication, while case 1b uses the Winograd approach as given in (2.5)-(2.8). Case 1c uses only the diagonal elements of the optimum $N \times N$ A_o matrix. From its ϵ^2, we see that a scalar filter (i.e., A set to a diagonal matrix) obtained from the diagonal components of A_o is quite inferior. It is because of this result that we are motivated to consider the pre-filtering operation T and post-filtering operation S . In case 1d (as well as in cases 2c, 2d, 3b, 4c and 4d), the notation of $\{D_o\}_{(1,\frac{1}{2},\ldots,1/2^p)}$ means that, from the diagonal matrix $D_o=\{d_{ii}\}$ with $d_{Max} = \underset{i}{Max}\{d_{ii}\}$, we obtain another diagonal matrix with its i^{th} component consisting of the rounding off of d_{ii} to the nearest value of $1/2^n$, $1 \leq n \leq$, times d_{Max}. Thus, by factoring out d_{Max} from each diagonal term, then $\{D_o\}_{(1,\ldots,1/2^p)}$ consists only of terms of powers of (1/2). But using a binary representation, multiplication of $1/2^n$ involves only n right-shifts. For most computers, right-shifts are intrinsically simpler than multiplications. Furthermore, the multiplication of d_{Max} can be factored into an overall scaling constant and need not be performed (except for dynamic-range scaling purposes). Since d_{ii} is a constant for all i , ϵ^2 for cases 1c and 1d are identical.

In case 2, S' is taken to be the Discrete Cosine Transform (DCT). In case 2a , T is also taken to be the DCT. It is known that when T=S'=DCT, the ϵ^2 is quite close to that of ϵ^2_{Min} (see [2]). In case 2b, T is taken to be the DCT matrix quantized to just $\{\pm 1\}$ and then normalized by dividing by the square root of N . This T is denoted by QDCT and no multiplication is involved in using such a matrix operation. We note that the M associated with case 2b is less than that of case 2a. In some space and satellite applications where T is possible performed on board before transmission, there may be motivation to use the simpler T = QDCT rather than T = DCT. We note that the ϵ^2 for case 2c is only slightly larger than that for case 2a , but the ϵ^2 for case 2d is considerably larger than that for case 2b .

In cases 3a and 3b, T = S' are taken to be the Walsh-Hardarmard Transform (Hardarmard represent.). In case 3a, there are only N multiplications in the diagonal terms. In case 3b, due to rounding off to powers of $1/2^n$, there is no multiplication. The ϵ^2 of these two cases are only slightly larger than that of ϵ^2_{Min} in case 1a.

Case 4 is similar to case 2 except that the Haar Transform is used instead of the Discrete Cosine Transform. All four associated are reasonably close to, and only slightly larger than, that of ϵ^2_{Min} in case 1a.

IV. CONCLUSIONS

We have considered various techniques in reducing the multiplicative computational complexity of the extended generalized Wiener filter. The penalty associated with the slight increase of ϵ^2 for considerable reduction of M appears to be quite significant. Much additional study appears necessary before any very general results can be claimed. Clearly, the number of multiplications is only one factor in the analysis of the through-put data rate in a digital signal processor. Memory and execution time are clearly also governed by the complexity of the programming instructions. For example, the programming for case 1b is obviously more complicated than that for case 1a. Furthermore, in the study of any digital signal processor, we need to consider the effect of quantization noise and proper scaling of data to prevent saturation. In any case, we believe that there is still much which needs to be studied in the design of special-purpose hardware devices for signal-processing.

ACKNOWLEDGMENT

The author wishes to thank Mr. S. Reisenfeld and Mr. Y.T. Yan for several informative discussions.

REFERENCES

1. Pratt, W.K.: Generalized Wiener filtering computation techniques, *IEEE Trans.Computers* C-21 (1972) 636-641.
2. Ahmed, N., Natarajan, T., and Rao, K.R.: Discrete Cosine Transform, *IEEE Trans.Computers* C-23 (1974) 90-93.
3. Ahmed, N., and Rao, K.R.: *Orthogonal Transform for Digital Signal Processing,* Springer Verlag, 1975.
4. Jain, A.K.: A fast Karhunen-Loève transform for a class of random processes, *IEEE Trans.Commun.* COM-24 (1976) 1023-1029.
5. Winograd, S.: A New Algorithm for Inner Product, *IBM Research Report* RC-1943 (1967).

ROBUST ESTIMATION USING THE ROBBINS-MONRO STOCHASTIC APPROXIMATION ALGORITHM

EDWARD L. PRICE
Naval Surface Weapons Center
Dahlgren, Virginia 22448

V. DAVID VANDELINDE
Department of Electrical Engineering
The Johns Hopkins University
Baltimore, Maryland 21218

ABSTRACT

The problem of minmax estimation of a location parameter introduced by Huber is considered. It is shown that under general conditions there exists a solution which is a form of the Robbins-Monro stochastic approximation algorithm. This generalizes earlier work by Martin and Masreliez who have given stochastic approximation (SA)-estimate solutions for two particular cases. As with the M-estimate solutions given by Huber, the SA solutions are completely determined by the probability distribution function with least Fisher information in the distribution set used to model the observation errors.

1. INTRODUCTION

Robust estimation has been a topic of continued interest since Tukey[15,16] showed that the classical least squares method may suffer a serious loss of efficiency in the case of non-Gaussian errors. Tukey and others have developed a variety of estimation procedures which are robust in the sense that the efficiency does not change drastically for small deviations of the error distributions from a nominal model. Most of the work to date has dealt with the simplest class of estimation problems, estimating a parameter from a sequence of measurements corrupted by additive noise. A review of this work and a fairly complete reference list may be found in [4]. Statisticians generally refer to this as estimation of a location parameter. Although engineers may be inclined to consider this problem too simple or narrow in scope to be of general interest, it has many direct practical applications. Furthermore, a clear understanding of how to obtain robust estimates for this problem is necessary before the idea of robustness can be successfully applied to more complex situations (e.g. obtaining sequential robust estimates of the state of a dynamic system driven by a random process from a sequence of noisy measurements.)

Robust estimation of a location parameter is the problem addressed by this paper. Perhaps the most appealing formulation of this problem is based on a minmax concept introduced by Huber[2]. In this approach, information on the probability distribution function (pdf) of the observation errors is assumed to be incomplete. (This is nearly always true in a practical situation.) The information available is used to define a partial specification for the pdf of the errors. This partial specification in turn defines a set consisting of all pdf's which meet the specification. A solution is provided by an estimation procedure and a corresponding least favorable distribution in the set of pdf's which is minmax with respect to the asymptotic variance of the sequence of estimates. This provides a means by which the quality of information on the underlying uncertainty influences the structure of the estimation procedure. If information on the observation errors is relatively complete and reliable, the corresponding distribution set used to model this uncertainty will be small and the minmax estimation procedure will be near optimal for each individual pdf in the set. On the other hand, if descriptive information on the observation errors is poor and incomplete, a relatively large set will be chosen to model the uncertainty, and though the minmax estimation procedure will be considerably less than efficient for some pdf's in the set, it will have the best possible performance (in terms of asymptotic convergence rate) which can be guaranteed for an arbitrary pdf in the set.

2. PROBLEM STATEMENT

The objective is to estimate the parameter θ from a sequence $\{x_n\}$ of observations. The observations are of the form

$$x_n = \theta + v_n , \quad n \geq 1,$$

where the v_n are independently distributed random variables, each with pdf G which is symmetric about zero $(G(v) = 1 - G(-v^-))$. Available information on G is incomplete and so it is used to define a convex set C of symmetric pdf's, each with zero location parameter, to which G is confined. An estimate T is defined as a sequence $\{T_n\}$ of functions $T_n : R^n \to R$, where R is the real line. If F is in C and T is an estimate for which $T_n(\bar{x}_n) \to \theta$ almost surely or in probability and $n^{.5} T_n(\bar{x}_n)$ is asymptotically normal when the v_n are distributed as F $(\bar{x}_n = (x_1 \ldots\ldots x_n))$ denote the asymptotic variance V[T,F]. We seek a solution for the following problem.

Problem 1: Find a pair $[T_0, F_0]$ such that

$$V[T_0,F] \leq V[T_0,F_0] \leq V[T,F_0]$$

for F in C and T a regular, translation invariant estimate.

An estimate T defined by the sequence $\{T_n\}$ is regular if T_n is measurable for each n. The estimate T_0 minimizes the asymptotic variance maximized over C. If T_0 is used to estimate θ, the asymptotic variance of $T_n(\bar{x}_n)$ will be no greater than $V[T_0,F_0]$ regardless of which distribution from C happens to be G.

Let I be the functional which maps each pdf onto its Fisher information for location $(I(F) = \int (f'/f)^2 f \, dx$ if F has an absolutely continuous density f and $I(F) = \infty$ otherwise). Throughout the paper we shall use a prime to denote the derivative with respect to the argument and a lower case letter to indicate the density for the pdf having the corresponding upper case symbol. Let F_0 be a pdf which minimizes I over C, and put $\psi_0 = -f_0'/f_0$. Huber has shown that if the sequence of functions defining T_0 is given by

$$\sum_{i=1}^{n} \psi_0 (x_i - T_n(\bar{x}_n)) = 0, \qquad (1)$$

then, provided some regularity conditions on F_0 and C are met, the pair $[T_0, F_0]$ solves Problem 1. The estimate defined by (1) is the maximum likelihood estimate for the least favorable pdf F_0. Huber[2] called an estimate of this form an M-estimate. Note that the M-estimate solving Problem 1 is completely specified once the distribution F_0 of minimum Fisher information in C is known.

2. THE ROBBINS-MONRO STOCHASTIC ALGORITHM AS A ROBUST LOCATION ESTIMATE

In this section we show that in general, subject to regularity conditions, an estimate based on the Robbins-Monro stochastic approximation algorithm exists which solves Problem 1. This result corresponds to that given by Huber for M-estimates. A stochastic approximation (SA) estimate solving a special case of Problem 1 was first discovered by Martin [8]. He gave an SA solution for the case in which the distribution set C is in the ε-contaminated normal class. Later, he and Masreliez[9] found an SA solution for C in what they called the p-point class. Our results show how to obtain an SA solution for Problem 1 for a general distribution set C.

The SA-estimates have a significant computational advantage over M-estimates. The latter require solution of a non-linear equation (1) which gets increasingly burdensome as the number of observations increases. The SA-estimates are given by a simple recursive formula (the stochastic approximation algorithm) and do not require the explicit use of past observations in computing the current estimate. As pointed out by Martin and Masreliez [9], they are thus suitable for real-time applications. However, the M-estimates will, in general, converge more quickly.

Before establishing the existence of an SA estimate solving Problem 1 in the general case, it will be helpful to review the general solution procedure and to define some conventions and terminology which will permit a more concise statement of the results to follow.

There are three steps involved in finding a solution for Problem 1. These are essentially the same for both the M and SA-estimate solutions. Only the formulas defining the two estimates are different. The three solution steps are:

Step 1: Find a distribution F_0 which has minimum information in the set C.

Step 2: Check to see if the estimate T_0 defined from F_0 satisfies some available set of sufficient conditions for asymptotic normality. (General conditions sufficient for asymptotic normality have been given by Huber [3] for M-estimates and by Sacks [12] for SA-estimates.)

A positive result in Step 2 implies that $[T_0,F_0]$ solves Problem 1 over the set

$$C^* = \{ F \in C: I(F) < \infty \} \tag{2}$$

Step 3: If the result of Step 2 is affirmative, check to see if the solution can be extended from C* to all of C.

This solution procedure has been established by Huber [2] for M-estimates. We shall show that it holds as well for SA-estimates. That a positive result in Step 2 implies $[T_0,F_0]$ solves Problem 1 over C* for T_0 an SA-estimate is shown in theorem 1. Theorems 2 and 3 give conditions which are sufficient for extending the solution from C* to all of C. These conditions apply to M-estimates as well.

Hereafter, we shall use $T^M(T^{SA})$ to represent an M(SA)-estimate. If F_0 minimizes Fisher information on the set C, we define T_0^{SA} to be the sequence generated by the recursive formula

$$T_{n+1}(\overline{x}_{n+1}) = T_n(\overline{x}_n) + \psi_0(x_{n+1} - T_n(\overline{x}_n))/[(n+1)I(F_0)], \quad T_1(x_1) = x_1, \tag{3}$$

where $\overline{x}_n$ and ψ_0 are as previously defined. T_0^M is defined by (1). For each F in C, put

$$v_F = E_F\,[\psi_0^2(x-\theta)],$$

$$v_0 = E_{F_0}[\psi_0^2\,(x-\theta)], \text{ and } m_F(\alpha) = E_F[-\psi_0(x-\alpha)].$$

If the sufficient conditions for asymptotic normality given by Huber [3] are satisfied by ψ_0 and F , then

$$V[T_0^{M},F] = \frac{v_F}{[m'_F(\theta)]^2} \tag{4}$$

If ψ_0 and F meet the corresponding conditions given by Sacks [12], then

$$V[T_0^{SA}, F] = \frac{v_F}{v_0[2m_F'(\theta) - v_0]} \quad . \tag{5}$$

In the following we use T_0 in statements which apply to both T_0^{M} and T_0^{SA}. Since only translation invariant estimates are considered, there is no loss in generality from the assumption $\theta=0$ which is made for convenience through the rest of the paper. If the sequence $\{n^{.5}T_n\}$ corresponding to T_0 is asymptotically normal when the observation error variables v_n are distributed as F, we say that T_0 is asymptotically normal at F.

The first theorem gives conditions under which an SA-estimate solves Problem 1 over a set which contains only pdf's which have finite Fisher information.

theorem 1: Suppose that C is a convex set of pdf's with C* non-empty, and that F_0 minimizes I over C. If

(i) T_0^{SA} is consistent and asymptotically normal with variance given by (5), and

(ii) $m_F'(\theta) = -\int\psi_0(x)f'(x+\theta)\,dx$ at each F in C*,

then $[T_0^{SA},F_0]$ solves Problem 1 over C*.

Proof: A direct substitution in (5) yields $V[T_0^{SA}, F_0] = 1/I(F_0)$, the Cramér-Rao lower bound for F_0. It follows that

$$V[T_0^{SA},F_0] \le V[T,F_0] \tag{6}$$

for all regular, translation invariant estimates T. From conditions (i) and (ii) we obtain

$$V[T_0^{SA},F] = v_F/v_0[\int(-2\psi_0(f'-f_0')-\psi_0^2(f-f_0))dx+v_F], \tag{7}$$

F in C*. But since F_0 has minimum Fisher information over C*, theorem 2 of [2] proves that $\int[-2\psi_0(f'-f_0') - \psi_0^2(f-f_0)]\,dx \ge 0$, F in C*.

This inequality, through (7), implies that

$$V[T_0^{SA},F] \le 1/v_0 = V[T_0^{SA}, F_0],\ \text{F in C*}. \tag{8}$$

The relations (6) and (8) establish the proof.

Condition (ii) allows Huber's differentiability results for I to be applied which makes the proof a simple matter. Some conditions on ψ_0 which imply condition (ii) are

(a) ψ_0 is absolutely continuous and ψ_0' is uniformly bounded a.e. on R with respect to Lebesgue measure, or the weaker condition,

(b) For each F in C* there is an F-integrable function h_F and a positive number ζ_F such that

$$|(\psi_0(x)/\alpha)\,[f(x+\alpha) - f(x)]\,| \le h_F(x),\ \text{F a.e., when } |\alpha| < \zeta_F.$$

The stronger and more easily verified condition (a) is not restrictive. The distribution of least Fisher information in C is also the smoothest in a sense. Thus, if C is a relatively full set, ψ_0 will be well behaved and condition (a) will be easily satisfied.

4. EXTENDING THE MINMAX ESTIMATE TO LARGER DISTRIBUTION SETS

Now we turn our attention toward extending to all of C a solution for Problem 1, which has already been shown to hold over those distributions in C which have finite Fisher information. Conditions allowing this extension are given which apply to both M and SA-estimates. Since the second inequality in the saddle condition in Problem 1 is verified in obtaining a solution which holds over C*, to extend this solution to the whole set C only requires verification that the first inequality of the saddle condition holds over C. The two theorems to follow give conditions which are sufficient to establish this first inequality.

The next theorem gives a condition for checking to see if the distribution set C is contained in the largest set (the set of all pdf's which satisfy (9)) over which

$$V[T_0^{SA},F] \leq V[T_0^{SA},F_0].$$

If the answer is affirmative then also

$$V[T_0^{M},F] \leq V[T_0^{M},F_0] \text{ over C.}$$

theorem 2: Assume that F_0 has minimum Fisher information in C, $T_0^M(T_0^{SA})$ is asymptotically normal with mean θ and positive finite variance given by (4) ((5)), and $m'_F(\theta) = \int\psi'_0(x-\theta)dF(x)$ for each F in C. Then $[T_0^M,F_0]$ solves Problem 1 if ($[T_0^{SA},F_0]$ solves Problem 1 if and only if)

$$\int[2\psi'_0(x-\theta) - \psi^2_0(x-\theta)]d(F(x) - F_0(x)) \geq 0 \qquad (9)$$

for each F in C.

Proof: As in the proof of theorem 1, $V[T_0,F_0] \leq V[T,F_0]$ for any regular translation invariant estimate T. (Note that $V[T_0^M,F_0] = V[T_0^{SA},F_0]$.) Taking $\theta=0$, it follows from the hypothesis on $m'_F(\theta)$ that

$$\int\psi'_0 dF_0 = -\int\psi_0 f'_0 dx = \int\psi_0^2 dF_0 = v_0,$$

and from this equation that

$$2m'_F(0) - v_0 = \int(2\psi'_0 - \psi_0^2)d(F - F_0) + v_F.$$

Substituting the preceding equation into (5) gives

$$V[T_0^{SA},F_0] - V[T_0^{SA},F] = \frac{\int(2\psi'_0-\psi_0^2)d(F-F_0)}{v_0[2m'_F(0)-v_0]} . \qquad (10)$$

The hypothesis that $V[T_0^{SA},F]$ is positive and finite implies that the denominator in (10) is positive and hence $V[T_0,F_0] - V[T_0^{SA},F] \geq 0$ if and only if $\int(2\psi'_0-\psi^2_0)d(F-F_0) \geq 0$ which proves the part of the theorem regarding SA-estimates. It follows directly from (4) and (5) that

$$V[T_0^{SA},F] - V[T_0^{M},F] = \frac{V_F[m'_F(\theta)-v_0]^2}{v_0[2m'_F(\theta)-v_0][m'F(\theta)]^2} \qquad (11)$$

The hypothesis that both $V[T_0^{SA},F]$ and $V[T_0^M,F]$ are positive and finite implies that (11) is non-negative, and zero if and only if $m'_F(\theta) = v_0$. The rest of the proof now follows from the result just proved for SA-estimates.

While theorem 2 is appealing as a theoretical result, it would be of little practical value without some way to verify that the required inequality (9) holds other than checking each pdf in C individually. The next theorem gives conditions which are easily checked in most cases, and which guarantee that the inequality (9) holds over all of C. Thus it provides a practical way to utilize the result of theorem 2.

Remark: From equation (11) one obtains that $V[T_0^{M},F] \le V[T_0^{SA},F]$, with equality if and only if $m'_F(\theta) = v_0$. Thus, in general, the minmax M-estimate converges faster then the corresponding SA-estimate.

In the theorem that follows, we use the same symbol F to denote both a pdf and the corresponding measure on the Borel sets of $\overline{R}$, the extended real line.

theorem 3: Assume that (T_0,F_0) solves Problem 1 over C* and that the following conditions hold:

(i) $T_0^{M}(T_0^{SA})$ is consistent and asymptotically normal with positive finite variance given by (4) and ((5)).

(ii) ψ_0 is absolutely continuous (but not necessarily bounded) on $\overline{R}$.

(iii) There is a subset A of $\overline{R}$ such that F(A) = 0 for each F in C and ψ'_0 is bounded and upper semi-continuous on $\overline{R}$-A.

(iv) For each F in C there is a sequence $\{F_n\}$ from C* such that $\int\phi dF_n \to \int\phi dF$ for any bounded and continuous ϕ.

Then (T_0,F_0) solves Problem 1 over C.

Proof: First, we show that $m'_F(0) = \int\psi'_0 dF$ $(\theta = 0)$. For each $t \neq 0$ let

$$\phi_t(x) = \min\ \{M, \max(-M,[\psi_0(x) - \psi_0(x-t)]/t)\}\ ,$$

where M is a bound for $|\psi'_0|$ on $\overline{R}$-A. Then $|\phi_t| \le M$, ϕ_t is F integrable and $\phi_t \to \psi'_0$ F a.e. as $t \to 0$ for each F in C. By Lebesgue's dominated convergence theorem,

$$m'_F(0) = \frac{d}{dt}\int -\psi_0(x-t)dF(x)\ \Big|_{t=0} = \lim_{t\to 0}\int\phi_t dF = \int\psi'_0 dF$$

for each F in C.

To complete the proof we show that the inequality (9) holds over C and the result follows from theorem 2.

Since (T_0,F_0) is a solution over C* by hypothesis, theorem 2 implies that (9) holds over C*. Let B be the set of all closure points of $\overline{R}$ - A which are in A. For each x in B, let $B(i,x) = \{y:|y - x| < 1/i$ $y\ \varepsilon\ \overline{R} - A\}$. Define the function γ by the formula

$$\gamma = \begin{array}{ll} 2\psi'_0 - \psi_0^2 & \text{on } \overline{R} - A \\ \lim\limits_{i\to\infty}\ \sup\limits_{B(i,.)}\ (2\psi'_0 - \psi_0^2) & \text{on B} \\ 2M - \psi_0^2 & \text{on A-B.} \end{array}$$

Since ψ'_0 is upper semi-continuous (usc) and bounded on $\overline{R}$-A and ψ_0 is continuous, it follows that γ is usc and bounded above. Let $\{\gamma_m\}$ be a sequence of bounded continuous functions decreasing pointwise to γ (see Ash [1], p. 390, for proof of existence). Let F in C and let $\{F_n\}$ be a sequence from C* with the convergence property of condition (iv). By the monotone convergence theorem

$$\int\gamma_m dF \downarrow \int(2\psi'_0 - \psi_0^2)dF. \quad \text{Now for each m,}$$

$$\limsup_n \int(2\psi'_0 - \psi_0^2)dF_n \le \limsup_n \int\gamma_m dF_n = \int\gamma_m dF.$$

Taking the limit as $m \to \infty$ we get

$$\limsup_n \int(2\psi'_0 - \psi_0^2)dF_n \le \int(2\psi'_0 - \psi_0^2)dF.$$

Since (9) holds for each F_n the preceding inequality implies that it holds also for F.

The conditions of theorem 3 will typically be easy to check once the pdf F_0 is known. The available sufficient conditions for asymptotic normality are used to check (i). With F_0 known, a formula for ψ_0 is in hand. Thus (ii) is easily checked. The set A in condition (iii) is usually the set of points at which ψ'_0 does not exist. If this is the case, condition (iii) is satisfied if all pdf's placing positive mass on A are excluded from C. As an example, take the p-point class of Martin and Masreliez [9]. Here, for a given set in the class, ψ'_0 does not exist at two points ($\pm y_p$). Distributions placing positive mass at these points are excluded from the set. Condition (iv) is probably the most difficult to verify.

5. DENSENESS OF DISTRIBUTIONS OF FINITE FISHER INFORMATION

Our next objective is to show that condition (iv) of theorem 3 holds on distribution sets in three general classes. Two of these, the ε - G and ε-contaminated G classes, were introduced by Huber[2] and have been considered by other authors in various robustness studies[6,7,8,9,13]. The third is a class which we call the generalized moment constrained (GMC) class. This class includes as special cases the p-point class considered by Martin and Masreliez[9] and the class of sets consisting of all pdf's taking common values at a specified finite set of points considered by Huber[5]. The approach is topological. We give the set P of all pdf's on the extended real line $\overline{R}$ the weakest topology under which all the maps $F \to \int\phi dF$ are continuous, where ϕ is bounded and continuous. This corresponds to the weak* topology induced on P as a subset of NBV[$\overline{R}$], the normalized space of functions of bounded variation on $\overline{R}$ [14]. In all subsequent statements which mention topological properties (such as closure, continuity, denseness, etc.) associated with distribution sets, this is the topology which is implied.

Condition (iv) of theorem 3 is equivalent to the requirement that C* be dense in C. It is shown in [10] that the pdf's of finite Fisher information are dense in P. If C has non-empty interior relative to P and is contained in the closure of its interior, then it follows that C* is dense in C. This is the approach we take to establish that condition (iv) holds. Because the proofs of the results of this section are not central to the theme of this paper, they have been omitted.

The first considered is the ε-contaminated G class. Recall that the distribution set C is from the ε-contaminated G class if $C = \{ F \in P : F = (1-\varepsilon) G + \varepsilon H$ for some H in P$\}$ for some fixed positive ε less than one.

lemma 1: If C is in the ε-contaminated G class and G has finite Fisher information, then C* is dense in C.

Next we consider the ε-G class. For convenient reference, we note that if $0<\varepsilon<1$, and G is some pdf, then the set C is in the ε-G class if $C = \{F \in P : |F - G| \le \varepsilon$ on $\overline{R}\}$. The condition given for guaranteeing denseness of C* in the ε-G set C is less restrictive than the corresponding condition of lemma 1 for an ε-contaminated G set.

lemma 2: Suppose C is in the ε-G class, where G is a continuous pdf. Then C* is dense in C.

The result of lemma 2 does not hold for arbitrary G. For example, if G has a discontinuity of magnitude $\delta>0$ at x_0 and $\varepsilon<\delta/2$, then each pdf in the ε-G set is discontinuous at x_0 and hence has infinite Fisher information. C* is empty in this case.

Now we show that C* is dense in C for sets of the type

$$C = \{F \in P : \int s_n dF \le p_n, \ 1 \le n \le N\} \tag{12}$$

This class includes cases in which the distribution set is a general GMC set

$$C_a = \{F \in P : \int s_n dF = p_n, \ 1 \le n \le M, \text{ and } \int s_n dF \le p_n, \ M+1 \le n \le N\}$$

in the sense that if F_0 has minimum Fisher information in C_a, then there exist constants $q_n = \pm 1$, $1 \le n \le M$, such that F_0 also minimizes Fisher information over

$$C_b = \{F \in P : q_n \int s_n dF \le q_n p_n, \ 1 \le n \le M, \text{ and}$$

$$\int s_n dF \le p_n, \ M+1 \le n \le N\}.$$

Details are given in [10].

A set of this form conveniently models the error distribution when available information is in the form of moments, partial moments, or a histogram. This general class includes the p-point family of Martin and Masreliez[9] as well as the sets consisting of all pdf's having common values at a finite number of specified points considered by Huber[5].

lemma 3: Suppose C is defined by (12) and s_n is upper semi-continuous on $\overline{R}$ and bounded above on finite intervals, $1 \le n \le N$. If for some F_1 in C and for each n, $\int s_n dF_1 < p_n$, then C* is dense in C.

6. CONCLUSION

In this paper we have shown that Huber's robust estimation of location problem (Problem 1) in general has a solution which is a form of the Robbins-Monro stochastic approximation algorithm. This is in addition to the M-estimate solution established by Huber[2]. Thus we have generalized the earlier work of Martin[8] and Martin and Masreliez[9] who gave SA-estimate solutions for two particular cases of Problem 1. As with the M-estimate solutions, the SA-solutions are completely determined by the pdf F_0 with least Fisher information in the distribution set C used to model the observation errors. Once F_0 has been obtained, both the M and SA-estimates which are solutions are easily defined. Thus, a procedure for calculating the minimum Fisher information distribution for sets in the various classes which provide convenient error models is of crucial importance. We shall address this problem in a subsequent paper.

REFERENCES

[1] Ash, R. B., "Real Analysis and Probability," Academic Press, New York, 1972.

[2] Huber, P. J., "Robust Estimation of a Location Parameter," Ann. Math. Statist., vol. 35, pp. 73-101, 1964.

[3] Huber, P. J., "The Behavior of Maximum Likelihood Estimates Under Nonstandard Conditions," Proc. Fifth Berkeley Symp. Math. Statist., vol. 1, pp. 221-233, 1967.

[4] Huber, P. J., "The 1972 Wald Lecture, Robust Statistics: A Review," Ann. Math. Statist., vol. 43, pp. 1041-1067, 1972.

[5] Huber, P. J., "Fisher Information and Spline Interpolation," Ann. Statist., vol. 2, pp. 1029-1033, 1974.

[6] Kassam, S. A. and Thomas, J. B., "Asymptotically Robust Detection of a Known Signal in Contaminated Non-Gaussian Noise," IEEE Trans. Inform. Theory, vol. IT-22, pp. 22-26, 1976.

[7] Martin, R. D. and Schwartz, S. C., "Robust Detection of a Known Signal in Nearly Gaussian Noise," IEEE Trans. Inform. Theory, vol. IT-17, pp. 50 - 56, 1971.

[8] Martin, R. D., "Robust Estimation of Signal Amplitude," IEEE Trans. Inform. Theory, vol. IT-18, pp. 596-606, 1972.

[9] Martin, R. D. and Masreliez, C. J., "Robust Estimation via Stochastic Approximation," IEEE Trans. Inform. Theory, vol. IT-21, pp. 263-271, 1975.

[10] Price, E. L., "Robust Estimation by Stochastic Approximation and Error Models for Robust Procedures," Ph.D. Dissertation, The Johns Hopkins Univ., Baltimore, Md. 1977.

[11] Prokhorov, Y. V., "Convergence of Random Processes and Limit Theorems in Probability Theory," Theory of Prob. and its Appl., vol. 1, pp. 157-214, 1956.

[12] Sacks, J., "Asymptotic Distribution of Stochastic Approximation Procedures," Ann. Math. Statist., vol. 29, pp. 373-405, 1958.

[13] Sacks, J. and Ylvisaker, D., "A Note on Huber's Robust Estimation of a Location Parameter," Ann. Math. Statist., vol. 43, pp. 1068-1075, 1972.

[14] Taylor, A. E., "Introduction to Functional Analysis," Wiley, New York, 1958.

[15] Tukey, J. W. and Harris, T. E., Statistical Res. Group, Princeton Univ., Princeton, N. J., Memo. Rep. 31, 1949.

[16] Tukey, J. W., "A Survey of Sampling from Contaminated Distributions," in Contributions to Probability and Statistics, (Harold Hotelling Volume), Stanford, California: Stanford Univ. Press, 1960, p. 448.

ON ASYMPTOTICALLY PERFECT DISCRIMINATION WITH QUANTIZED DATA

D. Kazakos
SUNY at Buffalo
Amherst, New York 14260

SUMMARY

A sequence of observations $x^n = (x_o \ . \ . \ . \ x_n)$ is generated by one of two s-state, first order Markov chains, corresponding to each of two hypotheses, H_1, H_2. The state transition probabilities are:

$$\Pr[x_k=j \mid x_{k-1}=i, H_1] = p_{jik} \ , \ \Pr[x_k=j \mid x_{k-1}=i, H_2] = q_{jik} \quad i,j=1,\dots,s$$

Let: $P_k = \{p_{jik}\}$, $Q_k = \{q_{jik}\}$, $A_k = \{a_{jik}\}$

$$a_{jik} = (p_{jik} q_{jik})^{1/2}$$

$$\Pr[x_o=i \mid H_1] = p_{io} \ , \ \Pr[x_o=i \mid H_2] = q_{io}$$

If the Bayes decision rule between H_1, H_2 achieves a probability of error that goes to zero as $n \to \infty$, we say that asymptotically perfect detection (APD) is achieved. Due to the nonstationary statistics, APD is not always achieved. In the present paper, we show that we have APD if and only if

$$\| C_n \|_1 \to 0 \text{ as } n \to \infty$$

where C_n is an s-dimensional vector produced by the recursion:

$$C_{n+1} = A_n C_n \ , \ C_o = [(p_{1o} q_{1o})^{1/2}, \ . \ . \ . \ , (p_{so} q_{so})^{1/2}]^t$$

A neccesary condition for APD is, then:

$$\lim_{n\to\infty} \prod_{k=1}^{n} \min_{1 \leq i \leq s} \sum_{j=1}^{s} a_{jik} = 0$$

If P_k, Q_k are symmetric, a necessary and sufficient condition for APD is:

$$\lim_{n\to\infty} \prod_{k=1}^{n} |z_k| = 0$$

where z_k is the absolutely maximum eigenvalue of A_k.

The above results were motivated by discrimination analysis of s-level quantized data with dependent samples, appearing in statistical processing of Remote Sensing data.

Research supported by NSF Grant ENG 76-20295.

SOME RESULTS ON BINARY CONVOLUTIONAL CODES FOR A MULTIPLE ACCESS CHANNEL*

ROGER PETERSON AND DANIEL J. COSTELLO, JR.
Illinois Institute of Technology, Chicago, Illinois 60616

ABSTRACT

Binary convolutional code pairs are investigated for use on the two-user adder channel. Conditions on the code generator polynomials of two R=1/2 coders which insure unique decodability on the noiseless adder channel are derived. It is proven that the free distance of the code pair is equal to the free distance of the sum code and that no linear uniquely decodable code pair exists having a rate pair above time sharing.

I. INTRODUCTION

Consider the two-user communication system illustrated in Fig. 1. Assume that the source outputs are binary and also that binary output convolutional encoders are employed. The two-user adder channel being considered is characterized by the transition diagram of Fig. 2. The channel is memoryless and accepts two inputs (one from each encoder) each unit of time and emits a single output from the alphabet {0,1,2}. In the noiseless case the channel output is a normal arithmetic sum of the inputs. The primary difficulty in finding codes for this channel stems from the property that two distinct inputs, 01 and 10, result in identical channel outputs.

Block coding for the two-user adder channel has been investigated by Kasami and Lin (1), (2), and (3). One example of a two-user block code is given in Fig. 3 where user 1 is assigned two code words and user 2 is assigned three code words. The table shown gives the noiseless channel output as a function of the code word pair transmitted.

The two-user adder channel capacity region has been determined by Liao (4) and Slepian and Wolf (5). Their results are illustrated in Fig. 4. The dotted line indicates the rate pairs achievable by time sharing. In addition, both Liao and Slepian and Wolf have proved a block coding theorem and its converse showing that block codes exist which achieve communication with arbitrary probability of error at any rate pair within the capacity region.

Although some two-user block code results have been found, almost no results exist for two-user tree, trellis, or convolutional codes. Ohkubo (6) proposes to bound the error probability for terminated convolutional codes on the general two-user discrete memoryless channel, but the validity of his result is in question.** In this paper a number of results concerning binary convolutional codes for the two user adder channel will be presented. Interest in convolutional, trellis, and tree codes for this channel is aroused by the possibility that these codes will have the same advantages over block codes as in the single user case.

II. DEFINITIONS

In all cases sequences of symbols will be represented by polynomials in the delay operator D. For example, the user 1 source output sequence i_{10}, i_{11}, i_{12} $\cdots$ is represented by $I_1(D) = i_{10} + i_{11}D + i_{12}D^2 + \cdots$. The user being considered is indicated by the first subscript. Convolutional

---

* This work supported by the National Science Foundation under grant ENG-75-20068.

**We believe that Ohkubo has not considered all possible error events.

encoders will be defined by their generator polynomial matrix denoted $\underline{G_1}$ or $\underline{G_2}$. Forney (7) discusses general convolutional coding techniques and explains the notations used here in detail. The user 1 and 2 encoder output sequences will be denoted $\underline{X_1}$ and $\underline{X_2}$. Note that if encoder 1 is an N output encoder, $\underline{X_1}$ represents N polynomials in D, e.g., if N=3, $\underline{X_1} = [X_{11}(D), X_{12}(D), X_{13}(D)]$.

Denote the noiseless adder channel output resulting from the input sequence pair $(\underline{X_1}, \underline{X_2})$ by $\underline{Y}(\underline{X_1}, \underline{X_2})$. Note that more than one polynomial might be represented by $\underline{Y}$. In particular, for encoders with three outputs, $\underline{Y} = [Y_1(D), Y_2(D), Y_3(D)]$, where each $Y_n(D)$ is a polynomial over {0,1,2}.

Definition 1 (1): A pair of convolutional encoders (C_1, C_2) is said to be uniquely decodable if and only if every distinct pair of encoder output sequences $(\underline{X_1}, \underline{X_2})$ generates a distinct noiseless channel output sequence $\underline{Y}(\underline{X_1}, \underline{X_2})$. Two encoder output sequence pairs $(\underline{X_1}, \underline{X_2})$ and $(\underline{X_1'}, \underline{X_2'})$ are distinct if $\underline{X_1} \neq \underline{X_1'}$ and/or $\underline{X_2} \neq \underline{X_2'}$.

Definition 2 (1): Consider two channel output sequences $\underline{Y}$ and $\underline{Y}'$. Define the L-distance, denoted $|\underline{Y}, \underline{Y}'|^L$ between the two sequences by

$$|\underline{Y}, \underline{Y}'|^L = \sum_n \sum_\ell |y_{n\ell} - y'_{n\ell}|,$$

where n indicates the sum over all polynomials represented by $\underline{Y}$ and ℓ indicates the sum over all coefficients in a particular polynomial.

Definition 3: The L-free-distance, d_{f_L}, of a convolutional code pair (C_1, C_2) is defined by

$$d_{fL} = \min_{(\underline{X}_1, \underline{X}_2) \neq (\underline{X}_1', \underline{X}_2')} |\underline{Y}(\underline{X}_1, \underline{X}_2), \underline{Y}(\underline{X}_1', \underline{X}_2')|^L.$$

III. DECODING

One of the most serious difficulties with two-user block codes is the problem of decoding. A significant advantage of two-user convolutional, trellis, or tree codes is that the decoding techniques used in the single-user case apply directly to the two-user case.

Consider the convolutional code pair having the trellis diagrams shown in Fig. 5. Because the channel is defined in terms of input pairs, the decoder must consider pairs of paths through the single user trellises. That is, the a posteriori probabilities of single paths are not defined, however the a posteriori probabilities of path pairs are defined. This leads immediately to the concept of the two-user trellis.

The two-user trellis is defined such that each distinct path pair through the two single-user trellises corresponds to a unique path through the two-user trellis. Each branch of the two-user trellis corresponds to a pair of branches in the single user trellises, and each state of the two-user trellis corresponds to a pair of states in the single-user trellises. The two-user trellis corresponding to the code pair of Fig. 5 is shown in Fig. 6, where the darkened path corresponds to the darkened path pair of Fig. 5. The branches of Fig. 6 are labeled with the noiseless adder channel output corresponding to channel input symbols on the associated single-user trellis branch pair. For example, the second branch of the darkened path of Fig. 6 corresponds to the user 1 transmitted symbols 11 and user 2 transmitted symbols 01 (see Fig. 5). The noiseless channel output in the first time unit of this pair is therefore 1 (adder channel input pair is 0 and 1) and in the second time unit is 2 (adder channel input pair is 1 and 1).

It can be shown that, under certain conditions on channel transition probabilities, a maximum likelihood estimate of the transmitted path is that path through the two-user trellis closest to the received sequence in terms of L-distance. In cases where the L-distance metric is not optimum, the log-likelihood function must be used as the path metric. The decoding problem is then to find the path through the two-user trellis closest to the received sequence. This can be accomplished using the Viterbi algorithm. The first two decoding steps using the L-distance metric are shown in Fig. 6 where the received sequence is given below the trellis and path metrics are given in parentheses above the states.

The fundamental concept underlying two-user decoding is the definition of the two-user trellis. The technique is not limited to the two-user adder channel and can in fact be applied to any two-user discrete memoryless channel using the log-likelihood function as a path metric. In addition, a two-user tree can be defined analogously and the concept extended to sequential decoding. Finally, the concept is not limited to two users. An M-user trellis or tree can be defined to extend the technique to an arbitrary number of users.

IV. RESULTS

Lemma 1: The code pair generated by two binary convolutional encoders each having the same number of outputs is uniquely decodable if and only if the two encoders have no common non-zero output sequence.

Proof: Consider two distinct pairs of encoder output sequences $(\underline{X}_1, \underline{X}_2)$ and $(\underline{X}'_1, \underline{X}'_2)$. Suppose that $\underline{X}_1 + \underline{X}'_1 = \underline{X}_2 + \underline{X}'_2$ where + indicates the mod-2 sum of the corresponding binary polynomials. Since the encoders are linear, and since either $\underline{X}_1 \neq \underline{X}'_1$ and/or $\underline{X}_2 \neq \underline{X}'_2$, $\underline{X}_1 + \underline{X}'_1 = \underline{X}_2 + \underline{X}'_2$ implies that the two encoders have a common non-zero output sequence. This in turn implies that if the encoders have no common non-zero output sequence then $\underline{X}_1 + \underline{X}'_1 \neq \underline{X}_2 + \underline{X}'_2$ for any distinct pairs $(\underline{X}_1, \underline{X}_2)$ and $(\underline{X}'_1, \underline{X}'_2)$. $\underline{X}_1 + \underline{X}'_1 \neq \underline{X}_2 + \underline{X}'_2$ implies $\underline{X}_1 + \underline{X}_2 \neq \underline{X}'_1 + \underline{X}'_2$. Thus for at least one component of one polynomial, $x_{1n\ell} + x_{2n\ell} \neq x'_{1n\ell} + x'_{2n\ell}$. Then either $x_{1n\ell} = x_{2n\ell}$ and $x'_{1n\ell} \neq x'_{2n\ell}$ or $x_{1n\ell} \neq x_{2n\ell}$ and $x'_{1n\ell} = x'_{2n\ell}$. In the first case $y_{n\ell} = 0$ or 2 and $y'_{n\ell} = 1$ and in the second case $y_{n\ell} = 1$ and $y'_{n\ell} = 0$ or 2. In both cases $y_{n\ell} \neq y'_{n\ell}$, so that $\underline{Y} \neq \underline{Y}'$ where $\underline{Y} = \underline{Y}(\underline{X}_1, \underline{X}_2)$ and $\underline{Y}' = \underline{Y}(\underline{X}'_1, \underline{X}'_2)$, and the code is uniquely decodable.

Now suppose the code pair has a common non-zero output sequence denoted by $\underline{X}_c$. Since the encoders are linear each has the zero output sequence $\underline{0}$. Then two distinct pairs of encoder output sequences are $(\underline{0}, \underline{X}_c)$ and $(\underline{X}_c, \underline{0})$, and, since the channel is not sensitive to numbering of the users, $\underline{Y}(\underline{0}, \underline{X}_c) = \underline{Y}(\underline{X}_c, \underline{0})$ and the code pair is not uniquely decodable. This in turn implies that if the code is uniquely decodable then the two encoders have no common non-zero output sequence.

Theorem 1: A pair of single input double output binary convolutional encoders ($R_1 = R_2 = 1/2$) generates a uniquely decodable code pair if and only if $g_{11}(D)g_{22}(D) \neq g_{12}(D)g_{21}(D)$.

Proof: Assume that both convolutional encoders generate distinct output sequences for all distinct input sequences. Suppose $\underline{X}_1 + \underline{X}_1' = \underline{X}_2 + \underline{X}_2'$ for two distinct pairs of encoder input sequences (I_1, I_2) and (I_1', I_2'). Since each encoder has two outputs this implies

$$I_1(D)g_{11}(D) + I_1'(D)g_{11}(D) = I_2(D)g_{21}(D) + I_2'(D)g_{21}(D)$$

and

$$I_1(D)g_{12}(D) + I_1'(D)g_{12}(D) = I_2(D)g_{22}(D) + I_2'(D)g_{22}(D).$$

Equivalently,

$$\left[I_1(D) + I_1'(D)\right]g_{11}(D) = \left[I_2(D) + I_2'(D)\right]g_{21}(D)$$

and

$$\left[I_1(D) + I_1'(D)\right]g_{12}(D) = \left[I_2(D) + I_2'(D)\right]g_{22}(D).$$

Multiply the top expression by $g_{12}(D)$ and the bottom by $g_{11}(D)$ and add to obtain

$$\left[I_2(D) + I_2'(D)\right]\left[g_{21}(D)g_{12}(D) + g_{22}(D)g_{11}(D)\right] = 0.$$

Similarly,

$$\left[I_1(D) + I_1'(D)\right]\left[g_{11}(D)g_{22}(D) + g_{12}(D)g_{21}(D)\right] = 0.$$

Since the encoder input pairs are distinct, these imply that

$$g_{11}(D)g_{22}(D) = g_{12}(D)g_{21}(D).$$

Thus, if $\underline{X}_1' + \underline{X}_1 = \underline{X}_2 + \underline{X}_2'$ for two distinct encoder input pairs then $g_{11}(D)g_{22}(D) = g_{12}(D)g_{21}(D)$. This in turn implies that if $g_{11}(D)g_{22}(D) \neq g_{12}(D)g_{21}(D)$, then $\underline{X}_1 + \underline{X}_1' \neq \underline{X}_2 + \underline{X}_2'$ for any two distinct encoder input pairs, and, as in the proof of Lemma 1 the code pair is uniquely decodable.

Suppose now that $g_{11}(D)g_{22}(D) = g_{12}(D)g_{21}(D)$, and consider an arbitrary nonzero user 1 encoder input sequence and the specific user 2 encoder input sequence given by

$$I_2(D) = \frac{g_{12}(D)}{g_{22}(D)}\, I_1(D),$$

where it has been assumed that $g_{22}(D) \neq 0$. The user 2 encoder output is

$$\begin{aligned}\underline{X}_2 &= \left[I_2(D)g_{21}(D),\ I_2(D)g_{22}(D)\right] \\ &= \left[\frac{g_{12}(D)}{g_{22}(D)}\, I_1(D)g_{21}(D),\ \frac{g_{12}(D)}{g_{22}(D)}\, I_1(D)g_{22}(D)\right] \\ &= \left[I_1(D)g_{11}(D),\ I_1(D)g_{12}(D)\right] = \underline{X}_1.\end{aligned}$$

A common non-zero output sequence has been found so that, by Lemma 1, the code pair is not uniquely decodable. Thus $g_{11}(D)g_{22}(D) = g_{12}(D)g_{21}(D)$ implies that the code pair is not uniquely decodable so that if the code pair is uniquely decodable then $g_{11}(D)g_{22}(D) \neq g_{12}(D)g_{22}(D)$. This completes the proof of Theorem 1.

Before stating the next Lemma, one additional definition is necessary.

Definition: Consider two binary convolutional encoders each of which has the same number of outputs. A binary convolutional encoder whose input consists of all inputs from these two encoders and whose output is the mod-2 sum of corresponding outputs, i.e., $X_{11}(D) + X_{21}(D)$, $X_{12}(D) + X_{22}(D)$ etc., of the two encoders is called a mod-2 sum encoder. Note that the mod-2 sum encoder of a K_1 input N output and a K_2 input N output encoder is a $K_1 + K_2$ input N output encoder.

Lemma 2: The L-free-distance, d_{fL}, of a binary convolutional code pair is equal to the free distance, d_f, of the mod-2 sum code.

Proof: Let $\underline{X}_s$ and $\underline{X}'_s$ be any two distinct output sequences of the mod-2 sum code. The Hamming distance between these two sequences is at least d_f. Since $\underline{X}_s = \underline{X}_1 + \underline{X}_2$ and $\underline{X}'_s = \underline{X}'_1 + \underline{X}'_2$, in any position where $\underline{X}_s$ and $\underline{X}'_s$ differ either $x_{1n\ell} = x_{2n\ell}$ and $x'_{1n\ell} \neq x'_{2n\ell}$ or $x_{1n\ell} \neq x_{2n\ell}$ and $x'_{1n\ell} = x'_{2n\ell}$. In the first case, $y_{n\ell} = 0$ or 2 and $y'_{n\ell} = 1$ and $|y_{n\ell} - y'_{n\ell}| = 1$. In the second case, $y_{n\ell} = 1$ and $y'_{n\ell} = 0$ or 2 and $|y_{n\ell} - y'_{n\ell}| = 1$. Since $\underline{X}_s$ and $\underline{X}'_s$ differ in at least d_f positions, and since one unit of L-distance is accumulated in each of these positions, the L-free distance is at least d_f.

Since the sum code is linear and has free distance, d_f, there exists a sum code output sequence vector with Hamming weight d_f. Denote this sequence $\underline{X}_{s,H}$ and let the corresponding user 1 and user 2 encoder outputs be denoted $\underline{X}_{1,H}$ and $\underline{X}_{2,H}$, i.e., $\underline{X}_{s,H} = \underline{X}_{1,H} + \underline{X}_{2,H}$. $\underline{X}_{1,H}$ and $\underline{X}_{2,H}$ differ in exactly d_f places. Let $\underline{Y} = \underline{Y}(\underline{X}_{1,H}, \underline{0})$ and let $\underline{Y}' = \underline{Y}(\underline{0}, \underline{X}_{2,H})$. In all positions where $x_{1,Hn\ell} \neq x_{2,Hn\ell}$, either $y_{n\ell} = 0$ and $y'_{n\ell} = 1$ or $y_{n\ell} = 1$ and $y'_{n\ell} = 0$. In either case, $|y_{n\ell} - y'_{n\ell}| = 1$. In all places where $x_{1,Hn\ell} = x_{2,Hn\ell}$ either $y_{n\ell} = y'_{n\ell} = 0$ or $y_{n\ell} = y'_{n\ell} = 1$ and in both cases $|y_{n\ell} - y'_{n\ell}| = 0$. Thus, since $\underline{X}_{1,H}$ and $\underline{X}_{2,H}$ differ in exactly d_f positions, $|\underline{Y}, \underline{Y}'|^L = d_f$.

It has been shown that the L-free distance is at least d_f and in a single case $|\underline{Y}, \underline{Y}'|^L = d_f$, so that combining these two results, $d_{fL} = d_f$ and the lemma is proved.

Theorem 2: No convolutional code pair exists which is uniquely decodable and has a rate pair above time sharing.

Proof: A code pair has a rate pair above time sharing in the noiseless case if and only if $R_1 + R_2 > 1$, where $R_1 = K_1/N$ and $R_2 = K_2/N$ and where K_1 and K_2 are the number of inputs and N the number of outputs of the user 1 and user 2 codes. The mod-2 sum code is a linear binary output code having $K_1 + K_2$inputs and N outputs and rate $R_s = (K_1 + K_2)/N = R_1 + R_2$. At $R_1 + R_2 = R_s = 1$ all possible binary output sequences of the sum code are used. At rate pairs $R_1 + R_2 = R_s > 1$ at least one output sequence must be used twice, the free distance of the sum code is zero, and by Lemma 2 the

L-free distance of the code pair is zero. A code pair with L-free distance of zero is not uniquely decodable and the theorem is proved.

V. SUMMARY AND CONCLUSIONS

Binary convolutional code pairs for the two-user adder channel have been investigated. A decoding technique applicable to the general M-user discrete memoryless channel has been defined by noting that the usual Viterbi and sequential decoding algorithms can be used in an M-user trellis or tree.

It has been proved that unique decodability of a two-user rate 1/2 binary convolutional code pair is assured by choosing codes whose generator polynomials satisfy $g_{11}g_{22} \neq g_{12}g_{21}$. Finally, it has been proved that the L-free distance of a convolutional code pair is equal to the free distance of the mod-2 sum code and, as a consequence, that no uniquely decodable convolutional code pair exists having a rate pair above time sharing.

It is concluded that any trellis or tree code pair with L-free distance greater than 1 and rate pair above time sharing must contain at least one non-linear code, i.e., code search algorithms should be limited to codes of this type.

VI. ACKNOWLEDGEMENT

The authors wish to acknowledge T. Kasami and S. Lin (1), (2), and (3) whose work has been followed closely in some of the proofs presented in this paper.

REFERENCES

(1) T. Kasami and S. Lin, "Coding for a multiple-access channel", IEEE Transactions on Information Theory, Vol. IT-22, pp 129-137, March 1976.

(2) ________, "Bounds on the achievable rates of block coding for a memoryless multiple-access channel", submitted to IEEE Transactions on Information Theory.

(3) ________, "On a class of δ-decodable codes for a multiple access channel", submitted to IEEE Transactions on Information Theory, January 1977.

(4) Henry H. J. Liao, "Multiple-access channels", Ph.D. dissertation, Department of Electrical Engineering, University of Hawaii, Honolulu, 1972.

(5) D. Slepian and J. K. Wolf, "A coding theorem for multiple-access channels", Bell System Technical Journal, Vol. 51, pp 1037-1076, 1973.

(6) M. Ohkubo, "Convolutional tree codes for multiple-access channels", IEEE Transactions on Information Theory, Vol. IT-23, No. 3, pp 397-399, May 1977.

(7) G. D. Forney, Jr., "Convolutional Codes I: Algebraic Structure", IEEE Transactions on Information Theory, Vol. IT-16, pp 720-738, November 1970.

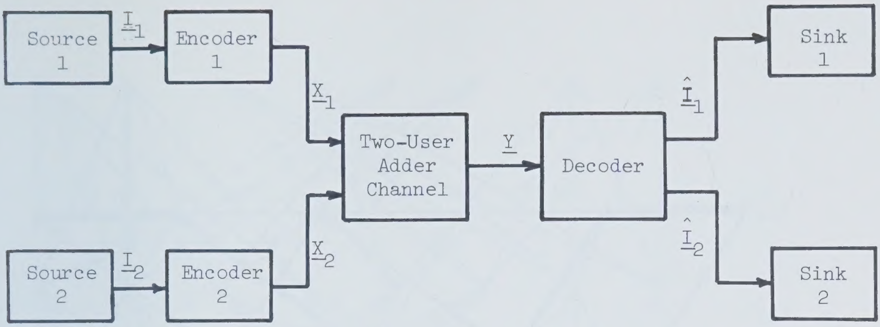
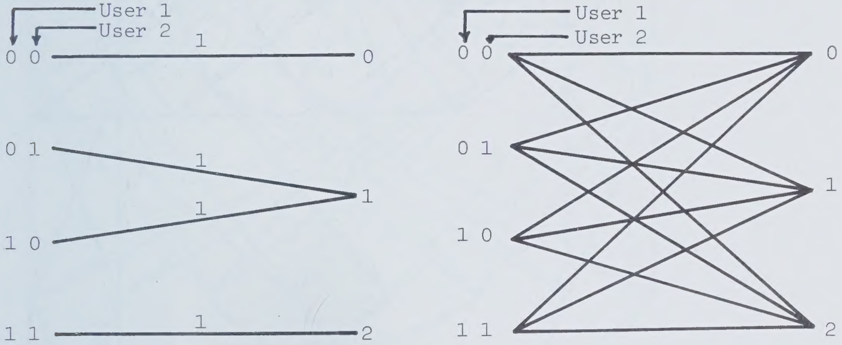

Source 1
$\underline{I}_1$
Encoder 1
$\underline{X}_1$
Source 2
$\underline{I}_2$
Encoder 2
$\underline{X}_2$
Two-User Adder Channel
$\underline{Y}$
Decoder
$\hat{\underline{I}}_1$
$\hat{\underline{I}}_2$
Sink 1
Sink 2

Fig. 1 Two-User Communication System


User 1
User 2
0 0
0 1
1 0
1 1
1
1
1
1
0
1
2
User 1
User 2
0 0
0 1
1 0
1 1
0
1
2

a) noiseless adder channel b) noisy adder channel

Fig. 2 Two-User Adder Channel Transition Diagrams

User 2 \ User 1	0 0	1 1
0 0	0 0	1 1
0 1	0 1	1 2
1 0	1 0	2 1

Fig. 3 Two-User Binary Code Pair

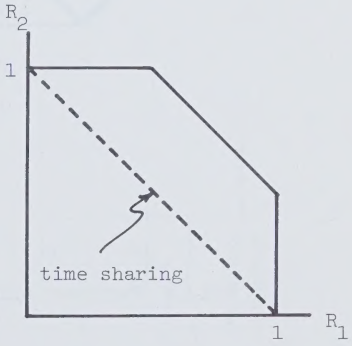
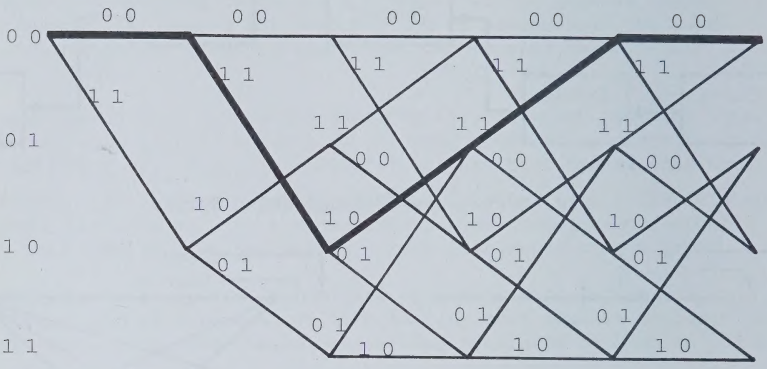
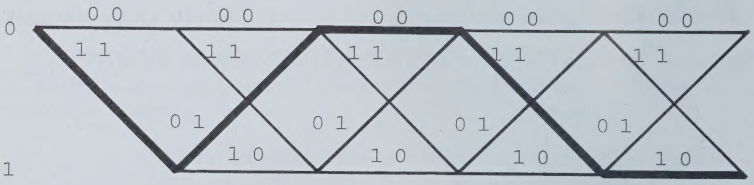


Fig. 4 Capacity Region of Noiseless Two-User Adder Channel


0 0
0 1
1 0
1 1
0 0
1 1
1 0
0 1
1 1
0 0
1 0
0 1

User 1 Code ($R_1 = 1/2$)


0
1
0 0
1 1
0 1
1 0

User 2 Code ($R_2 = 1/2$)

Fig. 5 Two-User Binary Convolutional Code Pair

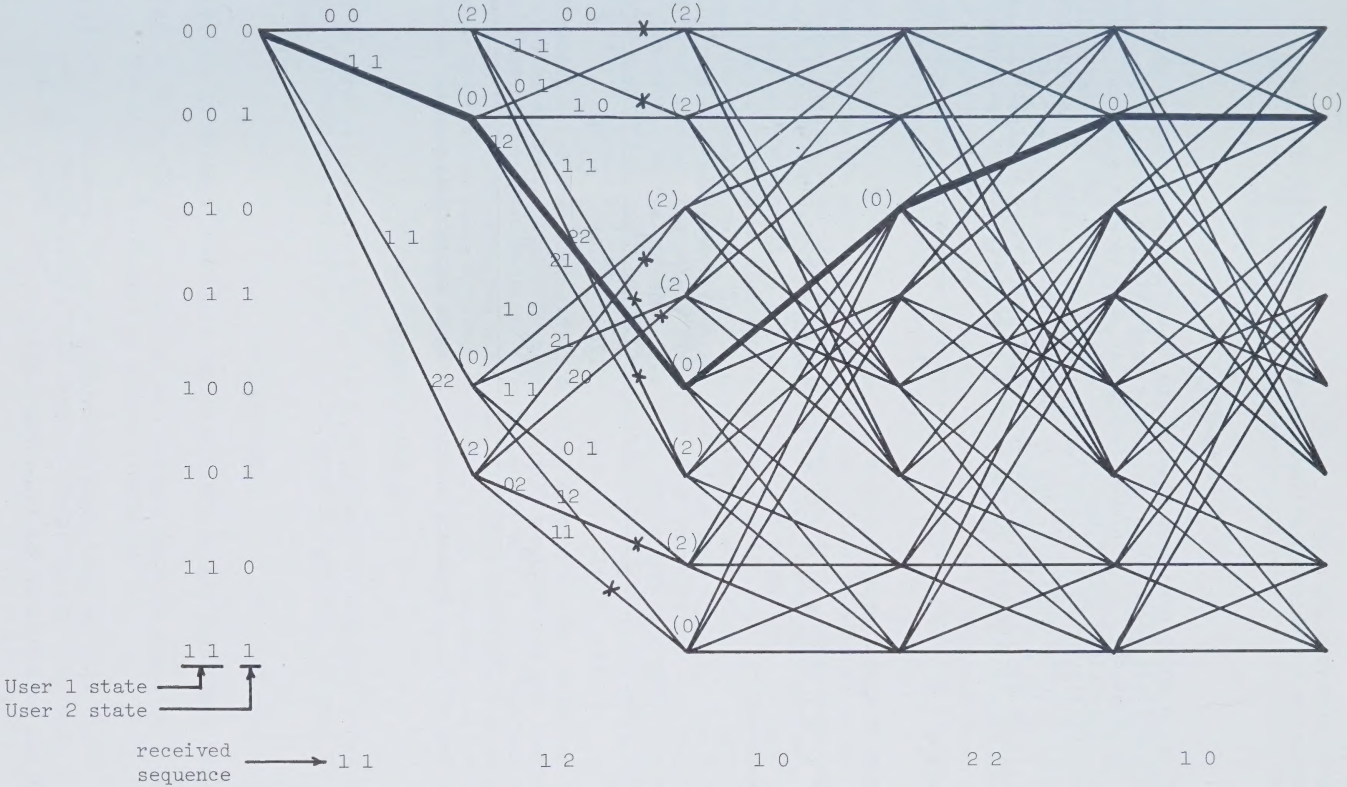

0 0 0
0 0 1
0 1 0
0 1 1
1 0 0
1 0 1
1 1 0
1 1 1
User 1 state
User 2 state
received sequence
1 1
1 2
1 0
2 2
1 0
0 0
1 1
(2)
(0)
0 1
1 0
1 1
12
22
21
20
02
11
(0)

Fig. 6 Two-User Trellis

A QUADRATIC CONVOLUTIONAL CODE*

JOHN P. ROBINSON and DANDU S. RAJU
Information Engineering, University of Iowa, Iowa City, Iowa 52242

SUMMARY

Most codes which have been studied are linear or group codes. We consider convolutional codes which are linear-quadratic functions of the information sequence. The minimum distance of such code is examined. An example of a quadratic convolutional code is given which has 7 occurrences of the minimum distance for each 8 for the best linear code. This code has block length 3, rate one-third, and an encoding span of 4 blocks. Let $x_o, x_1, \ldots, x_M$ be M + 1 binary variables corresponding to the current information bit and the bits in the encoder. The three encoder functions are:

$f_1 = x_o$,

$f_2 = x_o \oplus x_1 \oplus x_2 \oplus x_4 \oplus x_2x_3 \oplus x_2x_4 \oplus x_3x_4$,

$f_3 = x_o \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_2x_3 \oplus x_2x_4 \oplus x_3x_4$. Note that the code is

is systematic.

	Distance						
	7	8	9	10	11	12	13
Quadratic Example	112	352	240	128	144	32	16
Best Linear	128	256	320	256	0	0	64

Distance spectra for full set of tree pairs

---

*This work was supported by NSF Grant ENG75-08620

GENERALIZED MINIMUM DISTANCE DECODING OF COMPLETELY ORTHOGONALIZABLE BINARY CODES

CHRISTOPHER C. YU AND DANIEL J. COSTELLO, JR.
Department of Electrical Engineering, Illinois Institute of Technology
Chicago, Illinois 60616

SUMMARY

In most cases, the decoding of an error correcting code is achieved by means of a hard decision on each individual signal output from the channel and an algebraic algorithm based on the Hamming distance. Such decoding algorithms are generally simple to implement because the structural properties of the code can be utilized. A typical example is majority logic decoding [1]. However, as indicated by coding theorems, the hard decision decoder causes a significant loss in performance. This deficiency can be overcome by quantizing the outputs more finely in order to take advantage of likelihood information from the channel. One of the most important techniques for using such information in algebraic decoding is generalized minimum distance (G.M.D.) decoding [2].

In this paper, two G.M.D. decoding algorithms for completely orthogonalizable binary codes are presented. Let Q be the number of channel output quantization levels, J be the number of orthogonal parity check equations, and T be the number of trials [2] needed in a G.M.D. decoding algorithm. For any Q, only a single trial is needed in the first algorithm. If hard decisions are employed, such an algorithm suggests a modification of the conventional majority decoding rule. It is interesting to note that the modified majority rule will make fewer incorrect decisions than the usual one does when J is odd and all the weight $(J + 1)/2$ error patterns are considered. Both rules become equivalent when J is even. The second algorithm makes use of this modified majority decoding rule to achieve G.M.D. decoding with the number of trials $T = \log_2 Q$.

Simulations of the first G.M.D. decoding algorithm were performed for an A.W.G.N. channel. Both 2 level and 8 level quantization were condidered. The codes which were selected for simulation were a (7,3) maximum length code, a (15,7) BCH code, and a (21,11) difference set code. The results were compared to those obtained by Sundberg [3]. In each case, the new G.M.D. decoding algorithm shows a .5 db - 1 db improvement in performance.

ACKNOWLEDGEMENT

This work was supported by the National Science Foundation, under Grant ENG 75-20068, at the Illinois Institute of Technology, Chicago, Illinois 60616.

REFERENCES

[1] J. L. Massey, "Threshold Decoding", MIT Press, Cambridge, Mass., 1963.

[2] G. D. Forney, Jr., "Generalized Minimum Distance Decoding", IEEE Trans. Inform. Theory, pp 125-131, April 1966.

[3] C. E. Sundberg, "One-Step Majority-Logic Decoding with Symbol Reliability Information", IEEE Trans. Inform. Theory (corresp.), pp 236-242, March 1975.

APPLICATION OF THE R_0 CRITERION TO SPREAD SPECTRUM COMMUNICATIONS

H. de PEDRO and A. GIORDANO
GTE Sylvania, Eastern Division
77 "A" Street
Needham Heights, Mass. 02194

The "cut-off" rate criterion has been used extensively to obtain bounds on probability of error of coded digital systems and in the design of more efficient coding-modulation schemes. We present here an application of this narrowband interference or impulsive noise environments.

The R_0 criterion gives a bound on the probability of error of convolutionally coded systems and also a region of rates where operation at arbitrarily small error probability is possible. It has in fact been suggested that for a digital communication system, R_0 rather than the error probability is a more meaningful performance criterion. Previously the R_0 criterion has been applied to the coherent Gaussian channel. Here, we present results of the application of the R_0 criterion to a spread spectrum system operating in an atmospheric noise channel or in a channel with narrowband interference. The results indicate code rates that may be used to obtain good performance in the atmospheric noise channel as a function of the impulsiveness of the noise and in the narrowband interference channel as a function of the strength and bandwidth of the interference.

In order to combat narrowband interference or impulsive noise, either time domain excision (TDE) or frequency domain excision (FDE) algorithms may be used. In these cases the R_0 criterion provides a measure of performance of the algorithms and a base of comparison with coded systems.

On the basis of the R_0 criterion it is shown that in highly impulsive noise or strong narrowband interference, very low coding rates must be used to obtain good performance without the use of excision. The use of FDE or TDE results in a larger R_0 and allows much simpler coding implementations to be used. This implies that excision used in conjunction with simple coding algorithms can be an effective means of combatting impulsive noise and narrowband interference and represents an alternative to the use of low rate codes operating alone.

ERROR PERFORMANCE OF REDUCED-STATE VITERBI DECODING

JOSEPH CHAMMAS, DAVID HACCOUN, JEAN CONAN
Ecole Polytechnique de Montréal, Montréal, Canada

It is well known that Viterbi decoding (VD) is essentially a maximum likelihood decoding algorithm for convolutional codes over discrete memoryless channels. However because the computational decoding effort increases exponentially with the memory m of the code, Viterbi decoding is limited in practice to codes of constraint lengths $K=m+1$, not larger than 7 or 8. In order to reduce this computational load, consider a Viterbi decoder operating on a reduced-state trellis (RST), that is a trellis having 2^{μ} "pseudo-states" instead of the 2^{K-1} "true" states corresponding to a code of constraint length K, $K-1 \geq \mu$. In a reduced-state trellis with 2^{μ} "pseudo-states", each "pseudo-state" is the merging of the $2^{K-1-\mu} = 2^{\nu}$ true states, where for each of these true states the most recent μ bits are identical. Reduced-state Viterbi decoding (RSVD) may thus be seen as Viterbi decoding on a RST. In this paper we analyse the error performance of the RSVD and compare it with the usual Viterbi decoding of the same order of complexity.

Assuming the all-zero path is transmitted, an upper bound on the first-event error probability [1], $P_{\mu}(\varepsilon)$, for the RSVD over a binary symmetric channel (with cross-over probability p) is given by: $P_{\mu}(\varepsilon) \leqslant T_{\mu}(D)$, $D=2\sqrt{p(1-p)}$, where $T_{\mu}(D)$ is the transfer function of the modified state diagram connecting the true-zero state to the pseudo-zero state. Regarding the RST as generated by a time-varying convolutional code, the ensemble average sequence error probability, P_E, is found to be the same as for the Viterbi decoder with the same order of complexity, $K=\mu+1$. However, if the RSVD extends all paths tied at a given pseudo-state, then the RSVD will yield better performance at a cost of a slightly larger decoding effort. Now, since a correct pseudo-state does not necessarily imply a correct true state, then the bit error probability, P_B, of the RSVD will be larger than the usual VD.

Simulation results have shown that P_B is always greater with RSVD than with VD of the same order of complexity. Some improvements were obtained by variants of the RSVD algorithm: 1) All paths that are tied at a given "pseudo-state" are extended and 2) All paths whose metric is larger or equal than the smallest metric of a survivor are extended. The most promising variant is the "adaptive" RSVD whereby either the number of survivors or the value of μ depends on the state of the channel noise, leading to a variable computational effort and consequently to the need of an input buffer. Finally, it has been observed that most of the good codes for VD produce catastrophic error events when decoded by RSVD. This is due to the fact that following a first-error event an incorrect path will never merge again with the correct path. The adaptive scheme is very promising since it could yield an error performance comparable to that of VD at a smaller average (but variable) decoding effort [2]. Moreover, it can effectively break all catastrophic error propagation.

REFERENCES

[1] Viterbi, A.J., "Convolutional Codes and their performance in Communication Systems", IEEE Trans., Comm. Tech., Oct. 1971

[2] Conan, J. and Haccoun, D., "Reduced-state Viterbi decoding of convolutional codes", Proc. Allerton Conf. 1976.

TIME VARYING CONVOLUTIONAL CODES- A SYSTEM THEORY APPROACH

J.S. BEDI*, HARPREET SINGH, A.K. KAMAL
Electronics and Comm. Engg. Deptt.,U.O.R.,Roorkee(India)
*On study leave (Q.I.P.) from Thapar Engg. College,Patiala

ABSTRACT

A close relationship has been established between the time varying convolutional codes and time varying sequential machines. Conditions for the existance of feed forward inverses have been obtained for the time varying convolutional codes and a procedure is given to obtain such an inverse. Results of the well established system theory have been utilized to the benefit of coding theory.

INTRODUCTION

Massey and Sain [1] have established a close relationship between the coding theory, antomata and continuous system theory. The results obtained in [1] have been further exploited by Forney [2], Massey and Sain[3] Viterbi [4] and others to study the properties of convolutional codes and for giving a new look to coding theory. Costello [5] has shown that the lower bound on free distance for non systematic time varying convolutional codes (TVCC) exceeds the upper bound on systematic codes for all rates, thereby guaranting the existance of non systematic TVCC with larger free distance than any systematic code of the same rate and constraint length. The free distance has been bound with improvement in error probability with sequential decoding in [6]-[7] and with maximum likelihood decoding in [4]. This shows the superiority of TVCC over any systematic convolutional code of the same rate and constraint length. The object of this paper is to study the relationship between the TVCC and time varying linear sequential circuits (TVLSC) and to use the results of more advanced theory of linear time varying discrete systems to the benefit of coding theory.

TIME VARYING LINEAR SEQUENTIAL CIRCUITS

The encoder for a rate K/N TVCC is a time varying clocked linear sequential circuit [5]. A TVLSC is a network having a finite number K of input terminals and a finite number N of output terminals with the terminal signals ranging over a GF(q)[8]. Signals are assumed to appear at the terminals at discrete time instances t=0, 1, 2,.... The changes in the internal state of the network resulting from signals applied to the input terminals at a time t are assumed to be instataneous so that at a time t+1 the system is in a fixed internal state configuration. A TVLSC is formed by connecting together time varying scalar multipliers, adders and delayers over GF(q) in a well formed manner. A well formed network is defined as one for which all closed loops contain at least one delayer. A TVLSC having n delayers is refered to as an n dimensional TVLSC. For an n-dimensional TVLSC, the n delayers having been numbered as 1 through n, the state variables are written as $S_1(t)$, $S_2(t)$,....,$S_n(t)$ where $S_i(t)$ is the output of the i th delayer at time t.

A TVLSC is characterized by the following equations

$$\begin{aligned}\underline{S}(t+1) &= \underline{S}(t)\,\underline{A}(t) + \underline{u}(t)\,\underline{B}(t) \\ \underline{y}(t) &= \underline{S}(t)\,\underline{C}(t) + \underline{u}(t)\,\underline{D}(t)\end{aligned} \qquad (1)$$

where $t = 0, 1, 2, \ldots.$ is the time variable,

$\underline{u}(t)$ = K-dimensional input vector over GF(q)
$\underline{S}(t)$ = n-dimensional state vector over GF(q)
$\underline{y}(t)$ = N-dimensional output vector over GF(q)
$\underline{A}(t) = [a_{ij}(t)]$ nxn matrix over GF(q)
$\underline{B}(t) = [b_{ij}(t)]$ Kxn matrix over GF(q)
$\underline{C}(t) = [c_{ij}(t)]$ nxN matrix over GF(q)
$\underline{D}(t) = [d_{ij}(t)]$ KxN matrix over GF(q)

The response of a TVLSC (1) at time t to a unit impulse applied at time $\tau \le t$ is defined as

$$h_{\tau}(t) = \begin{cases} \underline{D}(\tau) & t=\tau \\ \underline{B}(\tau)\left[\prod_{j=\tau+1}^{t-1} \underline{A}(\tau)\right]\underline{C}(t) & t>\tau \end{cases} \qquad (2)$$

The t-zero-state response of the TVLSC is given in terms of the sequence $\underline{u}$, time t and the impulse responses as

$$g(0_t,\underline{u}) = \sum_{i=0}^{n} u_i\, h_{t+i}\,(t+n) \qquad (3)$$

It can be easily seen from (3) that the t-zero-state impulse response transfer function takes the form

$$\underline{H}_{\infty}(t) = \begin{bmatrix} \underline{D}(0) & \underline{B}(0)\underline{C}(1) & \underline{B}(0)\underline{A}(1)\underline{C}(2) & \ldots & \underline{B}(0)\prod_{1}^{n-1}\underline{A}(j)\underline{C}(n) \\ 0 & \underline{D}(1) & \underline{B}(1)\underline{C}(2) & \ldots & \underline{B}(1)\prod_{2}^{n-2}\underline{A}(j)\underline{C}(n) \\ 0 & 0 & \underline{D}(2) & \ldots & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \ldots & \underline{D}(n) \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \end{bmatrix} \qquad (4)$$

If the response of initial n pulses is considered, the semi-infinite matrix $\underline{H}_{\infty}(t)$ (4) becomes an $(n+1)K \times (n+1)N$ matrix.

$$\underline{H}(t) = \begin{bmatrix} \underline{D}(0) & \underline{B}(0)\underline{C}(1) & \underline{B}(0)\underline{A}(1)\underline{C}(2) & \cdots & \underline{B}(0)\mathop{\pi}\limits_{j=1}^{n-1}\underline{A}(j)\underline{C}(n) \\ 0 & \underline{D}(1) & \underline{B}(1)\underline{C}(2) & \cdots & \underline{B}(1)\mathop{\pi}\limits_{j=2}^{n-1}\underline{A}(j)\underline{C}(n) \\ 0 & 0 & \underline{D}(2) & \cdots & \underline{B}(2)\mathop{\pi}\limits_{j=3}^{n-1}\underline{A}(j)\underline{C}(n) \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & \underline{D}(n) \end{bmatrix} \qquad (5)$$

TIME VARYING CONVOLUTIONAL CODES

The encoding equations for a rate $R = K/N$ TVCC can be written as

$$\underline{y} = \underline{u}\,\underline{G}_\infty \qquad (6)$$

where $\underline{u} = (\underline{u}_0, \underline{u}_1, \underline{u}_2, \ldots.)$

$$= (u_0^{(1)}, \ldots u_0^{(K)}, u_1^{(1)}, \ldots, u_1^{(K)}, u_2^{(1)}, \ldots, u_2^{(K)}, \ldots) \qquad (7)$$

is the sequence of information digits,

$$\underline{y} = (\underline{y}_0, \underline{y}_1, \underline{y}_2, \ldots.)$$

$$= (y_0^{(1)}, \ldots, y_0^{(N)}, y_1^{(1)}, \ldots, y_1^{(N)}, y_2^{(1)}, \ldots, y_2^{(N)}, \ldots) \qquad (8)$$

is the sequence of transimitted digits or code word, and

$$\underline{G}_\infty = \begin{bmatrix} \underline{G}_0(0) & \underline{G}_1(1) & \cdots & \underline{G}_n(n) & & & \\ & \underline{G}_0(1) & \cdots & \underline{G}_{n-1}(n) & \underline{G}_n(n+1) & & \\ & & \ddots & & \ddots & \ddots & \\ & & & \underline{G}_0(n) & \underline{G}_1(n+1) & \cdots & \underline{G}_n(2n) \\ & & & \ddots & \ddots & & \ddots \end{bmatrix} \qquad (9)$$

is the semi-infinite generator matrix in which each $\underline{G}_h(t)$ is a KxN matrix, $0 \leq h \leq n$, $0 \leq t < \infty$. The elements of $\underline{u}$, $\underline{y}$, $\underline{G}_\infty$ are over GF(q). The matrix $\underline{G}$ for an (n+1)N, (n+1)K TVCC for the initial code vectors can be written as an (n+1)K x (n+1)N matrix [9]

$$\underline{G}(t) = \begin{bmatrix} \underline{G}_0(0) & \underline{G}_1(1) & \cdots & \underline{G}_n(n) \\ & \underline{G}_0(1) & \cdots & \underline{G}_{n+1}(n) \\ & & \ddots & \vdots \\ & & & \underline{G}_0(n) \end{bmatrix} \qquad (10)$$

where n is the number of memory elements.

The elementry condition for the code to be useful is that the rank of $\underline{G}$ is (n+1)K i.e. the code exhibits input functional reproducibility [1].

RELATION BETWEEN TVLSC AND TVCC

Following theorem establishes a relation between TVLSC and TVCC, the proof of which can be directly seen by comparison of eq.(3) with eq(6) and eq(5) with eq(10).

Theorem 1

The TVLSC (1) with S(0)=0 is an encoder for some TVCC if it has finite input memory and exhibits input functional reproducibility. Conversely, the encoder for a TVCC can always be taken as some TVLSC with S(0)=0.

Definitions

Periodic TVCC If T is the least integer such that

$$\underline{G}_h(t) = \underline{G}_h(t+T) \quad , \quad 0 \leq h \leq n \ , \ h \leq t < \infty \tag{11}$$

then the TVCC is defined as periodic TVCC with period T.

Eventually Periodic TVLSC A TVLSC is called eventually periodic if there exists positive integers τ and T such that for all $t \geq \tau$

$$\begin{aligned} \underline{A}(t) &= \underline{A}(t+T) \\ \underline{B}(t) &= \underline{B}(t+T) \\ \underline{C}(t) &= \underline{C}(t+T) \\ \underline{D}(t) &= \underline{D}(t+T) \end{aligned} \tag{12}$$

and the phase of TVLSC at time $t \geq \tau$ is given by

$$\Psi(t) = (t-\tau) \ (\text{mod } T) \qquad 0 \leq \Psi(t) < T$$

Strictly Periodic TVLSC An eventually periodic TVLSC is said to be strictly periodic TVLSC, if $\tau = 0$.

Corrolory 1

A strictly periodic TVLSC with S(0)=0 is an encoder for a periodic TVCC if and only if it has finite memory and exhibits input functional reproducibility. Conversely the encoder for a periodic TVCC can always be taken as a strictly periodic TVLSC with S(0)=0.

INVERSES

It has been shown [3] that to avoid 'catastrophic error propagation' the linear sequential circuits must have feedforward inverses, with or without delay. The same arguments applied to the TVCC necessitate to explore the existance of feedforward inverses with or without delay in the case of TVLSC. Forney [2] has given that every convolutional encoder is equivalent to a systematic encoder that may itself contain feedback, but which has a delay free feedback free inverse. By having a feedforward realization we mean that the transfer function characterizing the LSC should have a finite memory i.e. all the terms in these transfer functions are polynomials [3]. There exists a class [10] of time varying systems, in which the transition matrix of the impulse response function is given as matrix exponential.

In this class $\underline{A}(t)$ commutes with $\int_{t_o}^{t} A(\tau)d\tau$ and the impulse response function becomes

$$\underline{H}(t,\tau) = \underline{C}(t)e^{\underline{A}(t)(t-\tau)}\underline{B}(t) + \underline{D}(t)$$

$$= \underline{D}(t) + \underline{C}(t)\underline{B}(t) + \underline{C}(t)\underline{A}(t)\underline{B}(t)t+.... \qquad (13)$$

Since all the entries in (13) are polynomials in t with coefficients over GF(q), it can be treated as a class representing feed forward realization in TVLSC. The realization procedures for this class of time varying systems have been given by Bedi et al [11]-[12]. Moore and Silverman [13] have given condition of existance of delay free feedward inverses of the time-invariant case. Here we extend the results of [13] to the class (13) of time varying case.

Condition for Existance of Inverse

To ensure that a decoder exists, the impulse response matrix $\underline{H}(t,\tau)$ must be right invertible. Equivalently [3] there must exist a proper rational matrix R(t) such that

$$\underline{H}(t)\ \underline{R}(t) = \frac{1}{t^{L}} \cdot I_{K} \qquad (14)$$

$\underline{R}(t)$ is termed as an L-delay inverse of $\underline{H}(t)$ [3]. If there exists a proper rational matrix $\underline{R}(t)$ such that (14) holds with L=0, then $\underline{R}(t)$ is said to be delay free inverse of $\underline{H}(t)$. It is straight forward to say that $\underline{H}(t)$ has a delay free Inverse if and only if the rank of $\underline{D}(t)$ = K for all t.

Right Inverse Representation

A non singular output coordinate transformation is applied to (1) to put the system in a more convenient form.

Let $\underline{D}^*(t)$ be a right inverse of $\underline{D}(t)$ and let $\underline{N}(t)$ be a basis for the right null space of $\underline{D}$. Then KxK matrix

$$\underline{T}(t) = \begin{bmatrix} \underline{D}^*(t) \\ \underline{N}\ (t) \end{bmatrix} \qquad (15)$$

is non singular and define an output transformation $\underline{\omega}(t) = \underline{y}(t).\underline{T}(t)$ which yields the following system

$$\underline{S}(t+1) = \underline{S}(t)\ \underline{A}(t) + \underline{u}(t)\ \underline{B}(t) \qquad (16a)$$

$$\underline{\omega}_1(t) = \underline{S}(t).\underline{C}_1(t) + \underline{u}(t) \qquad (16b)$$

and $$\underline{\omega}_2(t) = \underline{S}(t)\underline{C}_2(t) \qquad (16c)$$

where $\underline{\omega}_1(t) = \underline{y}(t).\underline{D}^*(t)$; $\underline{\omega}_2(t) = \underline{y}(t).\underline{N}(t)$;
$\underline{C}_1(t) = \underline{C}(t).\underline{D}^*(t)$ and $\underline{C}_2(t) = \underline{C}(t).\underline{N}(t)$.

Let $\underline{H}_1(t)$ denote the impulse response function relating $\omega(t)$ to $\underline{u}(t)$. The Right inverse of $\underline{H}_1(t)$ can be characterized in terms of the following system, obtained by solving for $\underline{u}(t)$

in (16b) and substituting in (16a).

$$\underline{S}(t+1) = \underline{S}(t)\left[\,\underline{A}(t)-\underline{C}_1(t)\underline{B}(t)\,\right]+\underline{\omega}_1(t)\underline{B}(t) \tag{17a}$$

$$\underline{u}(t) = -\,\underline{S}(t)\underline{C}_1(t) + \underline{\omega}_1(t) \tag{17b}$$

$$\underline{\omega}_2(t) = \underline{S}(t)\underline{C}_2(t) \tag{17c}$$

Let $\underline{R}_1(t) = -\underline{C}_1(t).e^{-(\underline{A}-\underline{C}_1\underline{B})}.\underline{B}(t) + I_K$ denote the impulse response function relating $\underline{u}(t)$ to $\underline{\omega}_1(t)$ and let $\underline{L}_1(t)= \underline{C}_2(t).e^{-(\underline{A}-\underline{C}_1\underline{B})}.\underline{B}(t)$ denote the impulse response function relating $\underline{\omega}_2(t)$ to $\underline{\omega}_1(t)$ in 17.

It is clear that the NXK matrix $[R_1(t) : 0]$ is a right inverse of $\underline{H}_1(t)$ and it can be shown [13] that every right inverse of $\underline{H}_1(t)$ is written as

$$\underline{R}(t) = \left[\,\underline{R}_1(t):0\right] + \underline{Q}(t)\left[-\underline{L}_1(t) : I_{N-K}\right] \tag{18}$$

for some matrix $\underline{Q}(t)$.

Construction

Consider (17a)-(17c). If the pair $(\underline{C}_2,\underline{A}-\underline{C}_1\underline{B})$ is observable [14] it can be shown [15] that there exists a matrix K such that the matrix $(\underline{A}-\underline{BC}_1+\underline{K}\ \underline{C}_2)$ has characteristic polynomial t^n. Since $\underline{K}(-\underline{\omega}_2(t)+\ \underline{S}(t)\ \underline{C}_2(t)) = 0$, it follows from (17a-17c) that

$$\underline{S}(t+1) = \underline{S}(t)(\underline{A}-\underline{C}_1\underline{B}+\underline{K}\ \underline{C}_2) + \underline{\omega}(t)\left[\underline{B} : -\underline{K}\right] \tag{19a}$$

and $$\underline{u}(t) = \underline{S}(t)\ \underline{C}_1(t)+\underline{\omega}(t)\left[I_K : 0\right] \tag{19b}$$

is a feedforward right inverse system for $\underline{H}_1(t)$.

On the other hand, if the pair $(\underline{C}_2,A-\underline{C}_1\underline{B})$ is not observable, then it can be shown [16] that there exists a coordinate transformation $\underline{S}(t) = \underline{x}(t)\underline{T}(t)$ (20)

under which the system (17) takes the form,

$$\begin{bmatrix}\underline{x}_1(t+1)\\ \underline{x}_2(t+1)\end{bmatrix} = \begin{bmatrix}\underline{x}_1(t)\\ \underline{x}_2(t)\end{bmatrix}\begin{bmatrix}\underline{A}_{11} & 0\\ \underline{A}_{21} & \underline{A}_{22}\end{bmatrix} + \underline{\omega}_1(t)\begin{bmatrix}\underline{B}_1(t)\\ \underline{B}_2(t)\end{bmatrix} \tag{21a}$$

$$\underline{u}(t) = \begin{bmatrix}\underline{x}_1(t)\\ \underline{x}_2(t)\end{bmatrix}\left[\,-\underline{\bar{C}}_{11}\ -\ \underline{\bar{C}}_{12}\,\right] + \underline{\omega}_1(t) \tag{21b}$$

$$\underline{\omega}_2(t) = \underline{x}_1(t)\ \underline{C}_{21} \tag{21c}$$

where the pair $(\underline{C}_{21},\ \underline{A}_{11})$ is observable. If $\underline{A}_{22}$ has characteristic polynomial t^J then we may add to the right side of (21a) the quantity

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -\ \underline{\omega}_2(t)\underline{K}(t) + \underline{x}_1(t)\ \underline{K}(t)\ \underline{C}_{21}(t) \\ 0 \end{bmatrix} \qquad (22)$$

and produce a feed forward inverse.

It is to be noted that for computing the coordinate transformation (20), the observability matrix of [14] is to be used and then the procedure laid by [16] may be followed.

REFERENCES

[1] J.L. Massey and M.K. Sain, 'Codes, Automata, and continuous Systems : Explicit Interconnections'. IEEE Transactions on Automatic Control, Vol.AC-12, No.6, December, 1967,pp. 644-650.

[2] G.D. Forney,'Convolutional Codes I : Algebraic Structure'. IEEE Transactions on Information Theory, Vol.IT-16,No.6, November 1970, pp 720-738.

[3] J.L. Massey and M.K. Sain,'Inverses of Linear Sequential Circuits', IEEE Transactions on Computers, Vol. C-17,No.4, April 1968.

[4] A.J. Viterbi,'Convolutional Codes and their Performance in Communication Systems'. IEEE Trans. on Comm. Tech.,Vol. COM-19, pp 751-772, Oct. 1971.

[5] D.J. Costello,'Free Distance Bounds for Convolutional Codes'.IEEE Transactions on Information Theory, Vol.IT-20,No. IT-20, No.3, pp 356-365, May 1974.

[6] D.J. Costello,'A Construction Technique for Random-error-Correcting Convolutional Codes'. IEEE Transaction on Information Theory, Vol.IT-15, No.5, pp 631-636, Sept. 1969.

[7] J.L. Massey and D.J. Costello,'Non Systematic Convolutional Codes for Sequential Decoding in Space Applications'. IEEE Transactions on Communication Technology, Vol.COM-19, pp 806-813, Oct. 1971.

[8] D.R. Deuel, 'Time-varying Linear Sequential Machines', Ph.D. Thesis, Electrical Engg. Department, University of California, Berkeley, 1967.

[9] Shu Lin,'An Introduction to Error Correcting Codes',Prentice Hall, Inc; Englewood Cliffs, New Jersey, 1970.

[10] K. OGATA, 'Modern Control Engineering' Prentice Hall of India Pvt. Ltd., New Delhi, page 692, 1974.

[11] J.S. Bedi, H. Singh and A.K. Kamal,'On the Realization of a Class of Time Varying Impulse Response Matrix,'IEEE Proceedings, Vol.65, May 1977.

[12] J.S. Bedi, H Singh and A.K. Kamal,'Minimal Realization of a Class of Time Varying Dynamical Systems', To appear in

IEEE Proceedings.

[13] B.C. Moore and L.M. Silverman,'A new Characterization of Feedforward Delay-Free Inverses', IEEE Transaction on Information Theory, Vol.IT-19, No.1, pp 126-129, Jan.1973.

[14] M. Lal and H. Singh,'Computational Procedure for the Minimum Realization of Linear Time-Varying Systems',IEEE Trans. on Automatic Control, Vol.AC-16, No.1, pp.93-94, Feb.1971.

[15] S.K. Mitter and R. Foulkes,'Controllability and Pole Assignment for Discrete Time Linear Systems Defined Over Orbitrary Fields', SIAM.J.Contr.,Vol.9,pp 1-7, 1971.

[16] R.E. Kalman,'Mathematical Description of Linear Dynamical Systems'. SIAM.J.Contr.,Vol.1, No.2,pp.161-192, 1963.

GOAL ANALYSIS: IMPLEMENTATION OF CONTROL ACTIONS USING DECISION TABLES

ROBERT F. ERLANDSON
Electrical and Computer Engineering Department
Wayne State University
Detroit, Michigan 48202

ABSTRACT

The goal analysis methodology briefly described herein provides a powerful systems analysis tool which flows naturally into a decision table format. Since decision tables can be readily implemented into a finished program, the goal formulation allows for a natural flow from design to implementation.

Software costs are now the major costs associated with large computer-control system development. Hence the earlier in the design-implementation process that software errors can be detected the less costly they are to correct. Therefore this combination of design and analysis methodologies can facilitate the design effort and help reduce development costs.

INTRODUCTION

The need for a goal analysis methodology of the type being currently developed [1,2] arises from the need to better understand, analyze, control, and coordinate the increasing number of systems of organized complexity [3]. This type of system tends to be hierarchical, highly nonlinear (in terms of causal relationships), involve factors which are difficult if not impossible to quantify, and yet possess an organization and exhibit purposeful behavior. Purposeful behavior in this context means the behavior may be interpreted as directed to the attainment of a goal [4]; and while goal-directed behavior is a key attribute of many systems it is difficult to utilize without a formal definition of a goal.

Previous work in this area has dealt with the mathematical definition of a goal and the resultant properties and characteristics [1], as well as application of these ideas to different systems, e.g. modeling the goal-directed behavior of saccadic eye movement [5] and the modeling and analysis of various portions of telecommunications system (in particular, computer control and coordination functions within the telephone system.) [1,2]. The telecommunication applications were part of a larger system design effort and hence the modeling and analysis activity was preliminary to the design and implementation phases. Earlier reports discussed the mathematical properties and the modeling and analysis capabilities of the goal framework, but did not discuss another important aspect, i.e. the natural flow from modeling and analysis to design and implementation.

The existence of such a flow is important for large-scale computer controlled systems where software costs are now the major costs [6]. This type of flow enables one to move from very general control and coordination statements to a final detailed implementation in a series of steps. At each step analysis is possible so as to ensure design consistency and smooth transitions. The following sections will discuss the flow from goal formulation to decision table formulation to implementation.

GOALS

The process begins with the definition of a goal. The essential features of a goal are a performance measure, a performance criterion, a control action or decision set, and a value set which enables comparisons [1]. From this a goal may be defined as a function pair (p,h), where $p\colon X \to V$ and $h\colon X \to 2^V$ (2^V is the power set of V). The set X denotes a decision set, V is a value set and is ordered, e.g. V could be a partially ordered set (poset), a semi-order or totally ordered. The important thing is that the value set enables comparisons. The function p is a performance measure and establishes a value $p(x) \varepsilon V$, for all actions $x \varepsilon X$. The function h establishes a performance criterion for each control element. A goal is satisfied if and only if for $x \varepsilon X$, the associated performance value p(x) is in the acceptable or satisfactory set as specified by h(x), i.e. $p(x) \varepsilon h(x)$.

The basic definition of a goal can be naturally extended in many ways. Complex performance measures can be constructed by a point-wise extension of binary operations defined on V, to V^X[7], e.g., if x is a set and V is a poset (lattice, or totally ordered), then $V^X = \{p \mid p\colon X \to V\}$ and for p, $q \varepsilon V^X$, $x \varepsilon X$, and a binary operation *,

$$(p * q)(x) = p(x) * q(x).$$

Similarly one can combine logical and set theoretic operations to define composite goal structures. For example, if we specify two single goals, (p_1,h_1) and (p_2,h_2) where

$$\begin{array}{ll} p_1 : X_1 \to V_1 & p_2 : X_2 \to V_2 \\ h_1 : X_1 \to 2^{V_1} & h_2 : X_2 \to 2^{V_2} \end{array} \tag{1}$$

The composite goal $[(p_1,h_1) \;\&\; (p_2,h_2)]$ can be defined. This composite goal is satisfied by actions (x_1,x_2) where $x_1 \varepsilon X_1$ and $x_2 \varepsilon X_2$ if and only if

$$p_1(x_1) \varepsilon h_1(x_1) \quad \& \quad p_2(x_2) \varepsilon h_2(x_2) \tag{2}$$

i.e. x_1 and x_2 concurrently satisfy their respective component goals. Set theoretic operations can be used to extend the criterion set, e.g. if (p_1,h_1) and (p_2,h_2) are as in equation (1) then

$$(p_1,h_1 \cup h_2) \quad \& \quad (p_2,h_1 \cup h_2) \tag{3}$$

is also composite goal where for $x_1 \varepsilon X_1$ and $x_2 \varepsilon X_2$

$$(h_1 \cup h_2)(x_1,x_2) = h_1(x_1) \cup h_2(x_2) \tag{4}$$

and (x_1,x_2) satisfied the composite goal (3) if and only if

$$p_1(x_1) \varepsilon h_1(x_1) \cup h_2(x_2)$$

and

$$p_2(x_2) \varepsilon h_1(x_1) \cup h_2(x_2) \quad . \tag{5}$$

Thus we can build up quite complex structures from relatively simple building blocks. Often times in linear programming applications, an objective function is specified along with a set of constraint equations, such a formulation can be easily transformed into composite goals via set theoretic, lattice and logical operations (or vice versa)[8].

DECISION TABLES

These composite goals can then be used to model the goal-directed behavior of systems, as well as to analyze the goal structure's consistency [1,2]. As the design proceeds the goal structure tends to become more detailed and it is at this more detailed level that one can move from the goal format into a decision table format. This is possible when one can specify the goals in enough detail so as to satisfy roughly six conditions [9,11]. These include, defining specific boundaries for the problem, enumerating individual or elementary decisions, specifying necessary outcomes, specify a value for each outcome, defining decision parameters for each decision, and defining decision parameter tests and values [9].

A decision table is a structure for describing a set of related decision rules. The table specifies a number of conditions and rules, rules being different combinations of conditions. Each rule then specifies a given action. A decision table can be implemented by a programming language [9,10,11], either by hand coding or direct computer processing of the tables [10]; hence the natural flow "goal framework → decision table → implemented software." Note that at each stage of the process analysis tools are available and hence the consistency and utility of the design are under constant scrutiny. This enables one to catch and correct many logical errors before the system is implemented, this is much easier and cheaper than making corrections after implementation.

Figure 1 illustrates the essential features of a decision table. The condition stub of the table specifies a list of possible system conditions. The condition entry portion of the table lists rules R_1 to R_k. For rule R_1 condition 1 holds, condition 2 does not hold; the blank in condition 3 indicates a don't-care, etc. Therefore each rule specifies a particular set of systems conditions. The action stub lists all the actions which the decision system can make, and the action entry portion specifies for each rule what combination of actions are to be employed. For example, R_1 uses actions 1 and m.

	R_1	R_2	. . . .	R_k
CONDITION STUB	CONDITION ENTRY			
Condition 1	Y	N		N
Condition 2	N	Y		N
Condition 3				
.				
.				
.				
Condition n	N	Y		N
ACTION STUB	ACTION ENTRY			
Action 1	X			
Action 2		X		
.				
.				
.				
Action m	X			X

Figure 1. The general format of a decision table.

GOALS & DECISION TABLES

In general the goal specification for a system can be considered as specifying the condition stub and condition entry portion of the decision table. One condition would be the goal is satisfied, other conditions would follow from how the goal was not satisfied. For example, if a goal were to control the queue length to a system resource and the set of control actions

$$X = \{0, 50, 100\}$$

were levels of blocking, i.e. 0, 50 or 100 percent blocking of inputs depending on the queue length. The goal can be formalized as follows:

p - the performance measure is the relation between control level x, and queue length ℓ, $(x,\ell) \varepsilon p$;

h - the performance criterion could be defined as

$h(0) = [0, 6]$

$h(50) = [5, 10]$

$h(100) = [8, 10]$

where [0,6] is the acceptable number of inputs in queue for a resource when no blocking is in effect. When the number of waiting units exceeds 6, a new acceptable level goes into effect, i.e. between and including 5 to 10 units, etc. This information can be put into a decision table format as shown in Figure 2.

Note in Figure 2 that in condition 1 the goal is satisfied, i.e. $(0,\ell) \varepsilon p$ and $\ell \varepsilon [0,6]$ and hence no action is taken. The control is unchanged. If however the input increases and the units on queue exceed 6, then R_2 applies and action b) is taken, i.e. the control level is raised to 50% blocking of inputs. Given such a format the decision table can be implemented (or coded) by nested - IF statements which test the queue length and then specify the action.

	R_1	R_2	R_3	R_4	R_5	R_6	R_7
CONDITION STUB	CONDITION ENTRY						
1. $(0,\ell)$ such that $0 \leq \ell \leq 6$	Y	N	N	N	N	N	N
2. $(0,\ell)$ such that $\ell > 6$	N	Y	N	N	N	N	N
3. $(50,\ell)$ such that $5 \leq \ell \leq 10$	N	N	Y	N	N	N	N
4. $(50,\ell)$ such that $\ell < 5$	N	N	N	Y	N	N	N
5. $(50,\ell)$ such that $\ell > 10$	N	N	N	N	Y	N	N
6. $(100,\ell)$ such that $8 \leq \ell$	N	N	N	N	N	Y	N
7. $(100,\ell)$ such that $\ell < 8$	N	N	N	N	N	N	Y
ACTION STUB	ACTION ENTRY						
a. Control unchanged	X		X			X	
b. Control from 0 → 50		X					
c. Control from 50 → 100					X		
d. Control from 100 → 50							X
e. Control from 50 → 0				X			

Figure 2. A decision table corresponding to the performance criterion and performance measure equations.

CONCLUSION

Therefore in addition to providing system analysis and design aid, the goal formulation can be used to construct a control systems decision-making strategy and facilitate its implementation. In large software-based control systems such an aid is quite valuable. Software costs are now the major costs of most large computer system developments and they will apparently remain so [6]. The earlier in the design-implementation process that logical errors can be detected the less costly they are to correct, thus design and implementation techniques like the goal analysis aids presented herein can facilitate the design effort and help reduce development cost.

REFERENCES

1. R. F. Erlandson, "An Introduction to the Properties and Characteristics of Goals," Proc. of 1973 IEEE Conference on Decision and Control, p. 281-285, 1973.
2. R.F. Erlandson, "System Interaction Measures," IEEE Trans. on Systems, Man and Cybernetics, Vol. SMC-5, p. 116-124, Jan. 1975.
3. W. Weaver, "Science and Complexity," Amer. Scientist, Vol. 36, pp. 530-544, 1948.
4. A. Rosenblueth, N. Wiener and J. Bigelow, "Behavior, Purpose and Teleology," Philos. Sci., Vol. 10, pp. 10-24, 1973.
5. R. F. Erlandson, O.G. Fleming, "Uncertainty Sets Associated with Saccadic Eye Movements-Basis of Satisfaction Control," Vision Res., Vol. 14, pp. 481-486, 1974.
6. B. W. Boehm, Software and Its Impact: A Quantitative Assessment, Rand Corp., Dec. 1972, National Tech. Info. Service, AO-765 925.
7. J. A. Goguen, "L-Fuzzy Sets," J. of Math. Analysis & Applications, Vol. 18, pp. 145-147, 1967.
8. S. Eilon, "Goals and Constraints in Decision Making," Oper. Res. Quart. Vol. 23, pp. 3-15, 1972.
9. H. McDaniel (ed.), Applications of Decision Tables: A Reader, Brandon/Systems Press, Inc., N.Y., 1970.
10. ibid, An Introduction to Decision Logic Tables, John Wiley & Sons, Inc. N.Y., 1968.
11. S.L. Pollack, H.T. Hicks, and W.J. Harrison, Decision Tables: Theory and Practice, Wiley-Interscience, N.Y., 1971.

A MODEL OF HUMAN DECISION MAKING IN A FAULT DIAGNOSIS TASK

WILLIAM B. ROUSE
University of Illinois
Urbana, Illinois 61801

ABSTRACT

Utilizing elementary concepts from the theory of fuzzy sets as well as several non-fuzzy heuristics, a model is presented of human decision making in the task of trouble-shooting graphically displayed networks. The performance of the model is compared to the results of two experimental studies.

INTRODUCTION

In an earlier paper [1], it was suggested that an understanding of human decision making in fault diagnosis tasks will become increasingly important as systems become more automated and the human is required to fill the role of trouble-shooter. The two experimental investigations discussed in the referenced paper considered the diagnosis of faults in graphically displayed networks. The effects of network size (i.e., number of nodes), forced-pacing, computer aiding, and training were studied.

The purpose of this paper is to integrate this body of empirical evidence into a model of human decision making in fault diagnosis. The model to be presented utilizes elementary concepts from the theory of fuzzy sets [2] as well as several pattern-evoked, non-fuzzy heuristics. The performance of the model, in terms of the average number of tests until correct solution, is compared to human performance in the aforementioned experiments. Quantitatively, the comparison between model and human performance is quite favorable. Qualitatively, the psychological interpretation of the model's parameters is quite interesting, indicating the usefulness of the fuzzy set formulation.

A FAULT DIAGNOSIS TASK

The task considered in this paper is fault diagnosis of graphically displayed networks. An example is shown in Figure 1. This display was generated on a Tektronix 4010 by a DEC System 10. These networks operate as follows. A random number of arcs enter each node. Similarly, a random number of arcs emanate from each node. Nodes are devices that produce either a 1 or 0. Arcs emanating from a node carry the value produced by that node. A node will produce a 1 if:

1. All arcs entering the node carry values of 1.
2. The node has not failed.

If either of these two conditions are not satisfied, the node will produce a 0. (Nodes are like AND gates.) Thus, if a node fails, it will produce values of 0 on all the arcs emanating from it. Any nodes that are reached by these arcs will in turn produces values of 0. This process continues and the effects of a failure are thereby propagated throughout the network.

The specific networks that we have considered were constrained in the following ways. Looking at the nodes in Figure 1 as elements in a matrix, we will use the terminology of rows and columns to characterize the networks. Each network was square in that it had N rows and N columns of nodes. Nodes were numbered such that node $k=i+(j-1)N$ where i denotes the row and j denotes the column. Node k_1 in column j (where $k_1=i+(j-1)N$) could only connect to nodes k_2 in column j+1 (where $k_2=i+jN-2$, $i=+jN-1$, $i+jN$, $i+jN+1$, $i+jN+2$). Any of these five connections were equally likely to occur except when the value of k_2 did not denote a node in column j+1. In that case, the number of nodes accepting connections was decreased. The number of arcs emanating from a node was equally likely to be 1, 2, or 3. Only one node could fail within any particular problem. In the self-paced experiment, all nodes were equally likely to fail. In the forced-paced experiment, the output nodes (those in column N) could not fail.

A problem began with the display of a network with the outputs of column N shown. (See Figure 2.) Based on this evidence, the subject's task was to "test" arcs until the failed node was found. Figure 3 illustrates the manner in which arcs were tested. A * was displayed to indicate that subjects could choose an arc to test. They entered commands of the form k_1,k_2 and were then shown the value of the arc. (See upper left of Figure 3.) If they responded to the * with a simple "return", they were asked to designate the failed node. Then, they were given feedback about the correctness of their choice. And then, the next problem was displayed.

In the experiments to be discussed below, computer aiding was one of the experimental variables. The aiding algorithm is discussed in detail in reference 1. Succintly, the computer aid utilized the network topology and known outputs to eliminate nodes that were infeasible solutions. Also, it iteratively used the results of tests (chosen by the human) to further eliminate nodes. In this way, the "active" network iteratively became smaller and smaller.

REVIEW OF EXPERIMENTS

Two experiments were performed using the fault diagnosis task described above [1]. The first experiment was self-paced while the second experiment was forced-paced.

In the self-paced experiment, the size of the problem N was varied to include problem sizes of 3, 5, and 7 (9, 25, and 49 nodes, respectively). Since many of the interesting effects occurred with the large problems, the subsequent forced-paced experiment only considered problems of size 7. Replacing problem size as an independent variable was time allowed per problem, T. Values of T considered included 30, 60, and 90 seconds. The choice of these values was motivated by the results of the self-paced experiment which indicated that it would be difficult to consistently solve problems in 30 seconds while it would be relatively easy to solve problems in 90 seconds.

The variable T was integrated into the experimental scenario by adding a clock to the display. Subjects were allowed one revolution of the clock in which to solve the problem. The circumference of the clock was randomly chosen from the three values noted above. Subjects did not know how much time was allowed until each problem began. If subjects had not solved the problem by the end of the allowed time period, the problem disappeared and they were asked to designate the failed node.

As noted earlier, computer aiding was also an experimental variable. Further, training and transfer between displays was considered.

Experimental measures were average number of tests until correct solution and percent correct. The former measure was normalized for individual problem difficulty by subtracting the optimal (minimum) value from each measurement.

The self-paced experiment included 8 subjects while the forced-paced experiment included 12 subjects. Each subject solved 6 practice problems followed by three trials of 30 problems each. Thus, the results reviewed below are based on 20 different subjects and almost 2000 problem solutions.

Subjects' instructions in the self-paced experiment were to find the fault in the minimum number of tests while the subjects in the forced-paced experiment were instructed to find the fault within the time constraint.

REVIEW OF EXPERIMENTAL RESULTS

Results of the self-paced experiment indicated that human performance deviated from optimality as problem size increased and that computer aiding lessened this deviation. However, even without computer aiding, subjects performed much better than a "brute force" model which simply traces back from an arbitrarily selected 0 output. This result can be interpreted as meaning that subjects used the topology of the network (i.e.,structural knowledge) to a great extent as well as the knowledge of network outputs (i.e.,state knowledge).

Considering training and transfer in the self-paced experiment, transferring from aided to unaided displays had negative effects in terms of average number of tests until correct solution (for large problems) while such transfer had positive effects in terms of percent correct. While this result is interesting, the practical significance of the effect was by no means as large as the effects of problem size and computer aiding.

Results of the forced-paced experiment indicated that the time allowed per problem and computer aiding had significant effects on human performance. Training effects were only related to the time allowed per problem.

A particularly interesting result was that forced-paced subjects utilized strategies requiring many more tests than necessary. Even for T equals 90 seconds (self-paced subjects tended to choose a T of approximately 40 to 50 seconds), subjects only economized on tests to a slight extent. Thus, it appears that one of the effects of forced-pacing was that subjects chose to employ less information in their solution strategies.

A very important conclusion that was drawn from both experiments was that subjects had considerable difficulty in effectively utilizing the 1 outputs (i.e., indications that something had not failed). In fact, several subjects mentioned that a test which produced a 1 was nearly worthless. This inability to use information which does not directly support one's hypothesis has appeared in several other contexts reported in the literature [3].

The results of both experiments could be discussed in considerable more detail. However, the model presented in this paper primarily addresses the effects noted above. Thus, the model will now be discussed.

A MODEL

The task of fault diagnosis can be represented as an iterative two-phase process whereby one first separates the set of nodes into those that could possibly be the source of unacceptable outputs (i.e., the feasible set) and those that could not possibly be the source of unacceptable outputs (i.e., the infeasible set). One then selects a member of the feasible set, tests it, and integrates the results of the test into a new partitioning of nodes into feasible and infeasible sets. This process continues until the fault is isolated or, perhaps until one is unwilling to continue.

When a task starts, one has knowledge of the network topology as well as several outputs, some of which may be unacceptable (i.e., output=0) while others may be acceptable (i.e., output=1). Considering the feasible set, as long as one is sure that only one node has failed then, the failed node must be able to cause all of the 0 outputs. However, for complex networks, the human may have great difficulty in perceiving the relationship of all nodes to each 0 output. One can represent this phenomenon using the membership function concept from the theory of fuzzy sets [2]. Utilizing this approach, the membership of node i in the fuzzy set of feasible causes of the 0 output of node j will be defined as

$$\mu^0_{ij} = \begin{cases} 1 & x_{ij} \leq a_1 \\ \dfrac{1}{1+\left[\dfrac{x_{ij} - a_1}{a_2}\right]^2} & x_{ij} > a_1 \end{cases} \qquad (1)$$

where x_{ij} is the "psychological distance" between nodes i and j while a_1 and a_2 are free parameters.

Avoiding, for the moment, a definition of psychological distance, equation 1 can be utilized to find the membership of node i in the fuzzy set of possible causes of all known 0 outputs. Since node i must be capable of causing all 0 outputs, one desires the intersection of fuzzy sets of feasible solutions. This leads to an overall membership function for node i of

$$\mu_i^0 = \underset{j}{\mathrm{MIN}}\{\mu_{ij}^0\} \tag{2}$$

The 1 outputs also yield considerable information although, as noted above, humans appear to have difficulty in effectively utilizing this information. If acceptable performance of node i is a prerequisite to acceptable performance of node j and, one knows that node j is performing acceptably then, node i must belong to the infeasible set. The membership of component i in the fuzzy set of contributors to the 1 output of node j will be defined as

$$\mu_{ij}^1 = \begin{cases} 1 & x_{ij} \le b_1 \\ \dfrac{1}{1 + \left[\dfrac{x_{ij} - b_1}{b_2}\right]^2} & x_{ij} > b_1 \end{cases} \tag{3}$$

where x_{ij} is defined as above while b_1 and b_2 are free parameters.

Equation 3 can be used to find the membership of node i in the fuzzy set of contributors to each known 1 output. Since one is only interested in the possibility of node i contributing to any of the 1 outputs, the union of fuzzy sets of infeasible solutions is the appropriate operation. This leads to an overall membership of node i in the fuzzy set of infeasible solutions given by

$$\mu_i^1 = \underset{j}{\mathrm{MAX}}\{\mu_{ij}^1\} \tag{4}$$

Now, considering the overall fault diagnosis problem, one is interested in nodes with strong membership in the feasible set and weak membership in the infeasible set. The fuzzy set of such nodes can be defined by forming the intersection of the feasible set with the complement of the infeasible set. The membership of node i in the resulting set is given by

$$\mu_i = \mathrm{MIN}\{\mu_i^0,\ 1 - \mu_i^1\} \tag{5}$$

Knowing the above membership function for all nodes, one can then choose to test the node whose membership function is maximum, although alternative approaches are possible [2].

To test this model, one must choose values of a_1, a_2, b_1, b_2 as well as define psychological distance. After considering a variety of alternative definitions, it seems clear that psychological distance is related to both geographical distance and functional distance. Geographical distance between nodes is simply the ordinary, everyday distance as measured, for example, by a ruler. Functional distance expresses the functional relationship between nodes in terms of whether or not the acceptable operation of one node depends on the acceptable operation of another node. To illustrate these ideas, in many automobiles the geographical distance between the battery and brakelights is much greater then the geographical distance between the spare tire and the brakelights. However, one would not usually test the spare tire if the brakelights did not work.

The above ideas lead to the following procedure for calculating psychological distance. For a given output of 0 or 1, determine all of the nodes that are functionally related to the node whose output is of interest. (For the task utilized in this paper, functional relationships are completely defined by the reachability matrix of the network.) Order this set of nodes in terms of increasing geographical distance from the node of interest. Assign psychological distances as the rank order of each node. Thus, for example, the node that is geographically closest and functionally related to the node whose output is of interest will have a psychological distance of unity.

While this definition of psychological distance may seem somewhat contrived, it has one particularly important characteristic. The experimental results discussed above indicated that computer aiding substantially improves human fault diagnosis performance. This result certainly can not be attributed to the aiding changing the geographical distance among nodes. Nor can computer aiding increase the degree of functional relatedness of nodes. However, nodes do become "closer" because irrelevant information is iteratively eliminated. The rank ordering idea discussed above allows psychological distance to change as the computer eliminates nodes from consideration. Although, admittedly, the results presented here in no way prove that such a defintion of psychological distance is anything more than an operationally useful construct.

Thus far within this section of this paper, the fuzzy aspects of human decision making in fault diagnosis have been considered. From observing many experimental subjects and discussing their strategies with them, it became clear that there are also several non-fuzzy aspects of human fault diagnosis. These can be classified as pattern-evoked heuristics. The most prevalent heuristics were as follows:

H_1. If only one node functionally connects (not just reaches) to all (>1) nodes with known 0 outputs, then that node must also have a zero output.

H_2. If a node with known 0 output functionally reaches another node with known 0 output, the latter node can be eliminated from future consideration.

H_3. If a node functionally reaches a node with known 1 output, the former node can be eliminated from future consideration.

H_4. If faced with a long chain of equally likely feasible solutions, test a node in the middle of the chain.

H_5. If a node with known 0 output is such that all its inputs are known to be 1, then that node is the fault.

H_1 is a fairly general heuristic which interestingly (and not surprisingly) occasionally leads to mistakes. H_2 is specific to networks without feedback. H_3 can be accomplished using the membership function of equation 3 with large b_1. H_4 is the half-split heuristic which subjects appeared to only occasionally adopt. H_5 is rather obvious and was invented to give the fuzzy set algorithm a way of recognizing when the solution had been reached. Finally, it is important to note that this set of heuristics was compiled by observing subjects and the only justification of these choices is that they seem reasonable.

Thus, a set of five heuristics and a few elementary fuzzy set operations define a model of human decision making in fault diagnosis. This model was programmed such that, on each iteration, it first tried H_1, H_2, and H_5. (H_3 was effectively handled by the fuzzy set algorithm while H_4 was infrequently utilized by subjects.) If these heuristics did not locate the fault, the fuzzy algorithm was then utilized. If the fuzzy set algorithm produced more than a single node with maximum membership grade, the node geographically most distant from the known outputs was selected. (This last procedure yields, in effect, a sixth heuristic.)

COMPARISONS OF MODEL AND HUMAN PERFORMANCE

The model was utilized as a "subject" in the two experiments discussed above. The model solved 300 problems using the same conditions as the self-paced experiment and 100 problems using the same conditions as the forced-paced experiment.

While the model has four free parameters (a_1, a_2, b_1, and b_2), it was found that task performance was only slightly sensitive to a_1, fairly insensitive to a_2, and that several combinations of b_1 and b_2 could produce the same results. Thus, all the results presented here utilize $a_1=0.0$, $a_2=25.0$, and $b_2=0.0001$ while b_1 was varied to match model and human performance.

The insensitivity of model performance to a_1 and a_2 is fairly straightforward to explain. Assume that $b_1 = b_2 = 0.0$. Then, with $a_1 = 0.0$ and $a_2 > 0.0$, the model will always choose to test the least geographically distant node that can reach all known 0 outputs. (The minimization operation of equation 2 dictates this behavior.) If a_1 is increased significantly, the choice may no longer be the least geographically distant node, but the reachability requirement remains. Such a change in a_1 also usually, but not always, causes degraded performance. Non-zero b_1 and b_2 will effect the specific choice, but qualitatively the effect of a_1 and a_2 will not change.

Thus, the model in effect has one free parameter. This parameter reflects the model's ability to utilize the 1 outputs which, as will be seen, allows very interesting psychological interpretations of human performance.

Results are shown in Tables I and II. Considering the results for the unaided trials of the self-paced experiment, note that b_1 increases with problem size. For a problem size of 3, the b_1 is optimal. However, the human's ability to increase b_1 falls short of optimal as problem size increases since $b_1 = 20.0$ and $b_1 = 42.0$ are optimal for problem sizes of 5 and 7, respectively. Part of this b_1 deficiency may be attributed to subjects' choice since the subjects apparently could have used a larger b_1 for problem sizes of 5.

The results for the aided trials of the self-paced experiment are unaffected by the choice of b_1 since computer aiding optimally utilizes the 1 outputs to eliminate infeasible solutions. Thus, with computer aiding subjects could ignore the information provided by the 1 outputs. It appears that this is the major benefit of computer aiding.

The results for the forced-paced experiment shown in Table II are only for the effectively self-paced trials (T equals 90 seconds). For the trials with faster pacing (T equals 30 and 60 seconds), it was apparent that subjects often only completed problems that required few tests. Since the model presented here does not consider the time required to perform tests, the model will complete all problems including those requiring many tests. Thus, model and human performance would not be expected to compare favorably for trials with fast pacing.

The results in Table II indicate that subjects effectively ignored the information provided by the 1 outputs. As with the aided trials of the self-paced experiment, computer aiding eliminated this effect. Apparently, the forced-paced subjects used much less sophisticated solution strategies in that they only utilized the information provided by the 0 outputs.

Considering the results of both experiments, note that the definition of psychological distance proposed has allowed the effect of computer aiding to be illustrated without adjustments of model parameters. This is certainly one of the model's strong points.

CONCLUSIONS

The results presented in the previous section of this paper allow the following interpretation of human decision making in the fault diagnosis task considered.

In determining the cause of unacceptable outputs, humans were fairly good at using the topology of the network to form an initial set of feasible solutions from which to choose a test. However, they were ineffective at using the acceptable outputs and the network topology to update this feasible set by eliminating infeasible solutions. The benefit of computer aiding was that it minimized the effect of the human's inadequate strategy.

The fuzzy set model presented here has enabled a quite concise interpretation of human decision making in a fairly robust task. The model's parameters and its independent variable (psychological distance) allow a wealth of behavioral interpretations. The next phase of this research is the application of this formulation to more robust fault diagnosis situations. This should be especially interesting from a display design point of view.

ACKNOWLEDGEMENT

This research was supported by the National Aeronautics and Space Administration under NASA Ames Grant NSG-2119.

REFERENCES

1. W. B. Rouse, "Human Problem Solving Performance in a Fault Diagnosis Task", unpublished.

2. L. A. Zadeh, K. S. Fu, K. Tanaka, and M. Shimura, Eds., Fuzzy Sets and Their Application to Cognitive and Decision Processes, New York: Academic Press, 1975.

3. P. C. Wason and P. N. Johnson-Laird, The Psychology of Reasoning, Cambridge, MA: Harvard University Press, 1972.

	PROBLEM SIZE, N		
	3	5	7
Unaided Subjects	$\bar{t}$ = 1.34 σ_t = 0.26	$\bar{t}$ = 2.78 σ_t = 0.94	$\bar{t}$ = 3.35 σ_t = 1.34
Unaided Model ($a_1 = 0.0$, $a_2 = 25.0$, $b_2 = 0.0001$)	$\bar{t}$ = 1.34 σ_t = 0.10 $b_1 \geq 6.00$	$\bar{t}$ = 2.85 σ_t = 0.14 b_1 = 8.00	$\bar{t}$ = 3.43 σ_t = 0.18 b_1 = 18.0
Aided Subjects	$\bar{t}$ = 1.05 σ_t = 0.18	$\bar{t}$ = 1.96 σ_t = 0.71	$\bar{t}$ = 2.74 σ_t = 1.12
Aided Model ($a_1 = 0.0$, $a_2 = 25.0$, $b_1 = 0.0$, $b_2 = 0.0001$)	$\bar{t}$ = 1.18 σ_t = 0.10	$\bar{t}$ = 1.99 σ_t = 0.15	$\bar{t}$ = 2.95 σ_t = 0.20

Comparison of Model and Subjects
for Average Number of Tests Until Correct Solution (t)
for Self-Paced Experiments

TABLE I

	UNAIDED	AIDED
Subjects	$\bar{t} = 7.35$ $\sigma_t = 1.41$	$\bar{t} = 3.95$ $\sigma_t = 0.49$
Model ($a_1 = 0.0$, $a_2 = 25.0$, $b_1 = 0.0$, $b_2 = 0.0001$)	$\bar{t} = 7.13$ $\sigma_t = 0.42$	$\bar{t} = 3.46$ $\sigma_t = 0.17$

Comparison of Model and Subjects
for Average Number of Tests Until Correct Solution (t)
for Forced-Paced Experiment

TABLE II

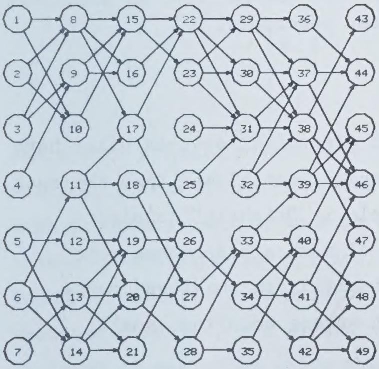

1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31 32 33 34 35
36 37 38 39 40 41 42
43 44 45 46 47 48 49

Fig. 1 An Example Network

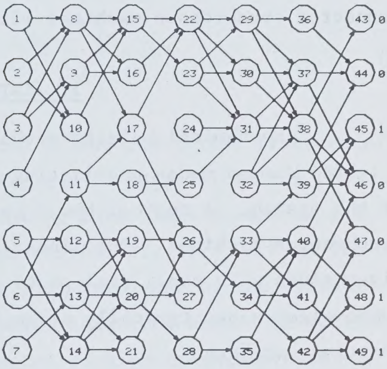
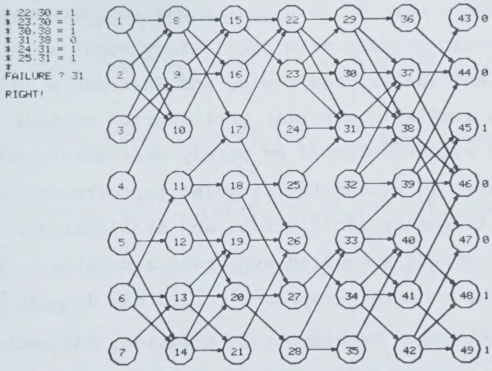

1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31 32 33 34 35
36 37 38 39 40 41 42
43 0
44 0
45 1
46 0
47 0
48 1
49 1

Fig. 2 An Example Network
With Outputs Shown



```
* 22,30 = 1
* 23,30 = 1
* 30,38 = 1
* 31,38 = 0
* 24,31 = 1
* 25.31 = 1
*
FAILURE ? 31

RIGHT!
```

1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31 32 33 34 35
36 37 38 39 40 41 42
43 0
44 0
45 1
46 0
47 0
48 1
49 1

Fig. 3 An Example Solution

MULTIPLE TIME SCALES FOR POWER SYSTEM ANALYSIS*

J. J. ALLEMONG and J. H. CHOW
Decision and Control Laboratory
Coordinated Science Laboratory
University of Illinois
Urbana, Illinois 61801

ABSTRACT

A technique to analyze systems with both high and low frequency oscillations is presented. The original system is decomposed into two lower order subsystems, one containing only the slowly varying part and the other containing only the fast oscillatory part. A system of 3 interconnected synchronous machines illustrates the procedure.

1. INTRODUCTION

Mechanical, electromechanical, and electrical systems often have slightly damped modes oscillating at frequencies much higher than the rest of the system. A typical power system example is "tightly" coupled synchronous machines. The linearized model of such a system has some eigenvalues with small real parts and relatively large imaginary parts. These eigenvalues typically arise because of strong coupling, small inertias, or both.

In well-behaved systems, the amplitudes of high frequency oscillations are small and their effect negligible. However, the analysis and design methods must take these potentially troublesome modes into account. These systems are numerically stiff; hence, sophisticated integration routines would have to be used. A way to circumvent this difficulty is to treat systems with oscillatory modes as singularly perturbed systems and analyze their slow and fast parts in different time scales. Presently available singular perturbation methods [1] do not permit the case of slightly damped or purely oscillatory modes. It is the purpose of this paper to extend the singular perturbation approach to systems with high frequency oscillations and to illustrate the methodology with an example of strongly coupled synchronous machines. Our results are in the spirit of the coherency approach [2] to the dynamic behavior of interconnected synchronous machines. In addition, our methodology extends

---

*This research was supported by the U. S. Energy Research and Development Administration, Electric Energy System Division, Contract No. EC-77-C-05-5566.

long term dynamics techniques [3] by reintroducing high frequency inter-machine oscillations as a correction to slow inter-area oscillations.

2. MODELING OF SYSTEMS WITH HIGH FREQUENCY OSCILLATIONS

Systems governed by physical laws such as Newton's law and Kirchhoff's law can be represented as second order matrix differential equations

$$\ddot{s} + Qs = 0, \quad \dot{s}(t_o) = \dot{s}_o, \quad s(t_o) = s_o \tag{1}$$

where s is an r×1 vector and Q is an r×r matrix. We note that no damping is present in (1). A more general derivation which includes damping is described in [4]. We assume that the system (1) is in the form

$$s = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \qquad Q = \begin{bmatrix} Q_1 & Q_2/\mu^2 \\ Q_3 & Q_4/\mu^2 \end{bmatrix} \tag{2}$$

where μ is a small positive parameter which arises due to the presence of stiff couplings or small masses or inertias and is responsible for the high frequency oscillations in (1). Isolating this small parameter and indeed identifying the physical quantities responsible for the high frequency oscillations at present requires a good knowledge of the system. Whenever possible, the governing differential equations should be written in terms of variables which make the presence of high frequency oscillations more readily apparent.

A very illuminating example of a system in the form of (1) is the three machine power system shown in Figure 1. Using the constant voltage behind transient reactance model [5], the following differential equations for this system may be written

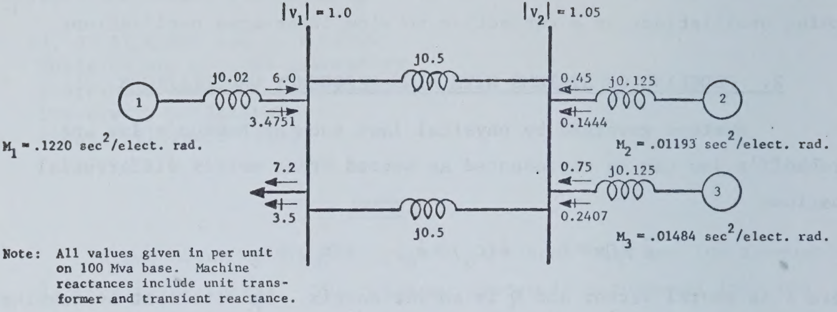
$$M_1\ddot{\delta}_1 = P_{in_1} - V_1'^2 Y_{11}\cos\theta_{11} - V_1'V_2'Y_{12}\cos(\theta_{12}+\delta_2-\delta_1) - V_1'V_3'Y_{13}\cos(\theta_{13}+\delta_3-\delta_1) \tag{3a}$$

$$M_2\ddot{\delta}_2 = P_{in_2} - V_2'^2 Y_{22}\cos\theta_{22} - V_1'V_2'Y_{12}\cos(\theta_{12}+\delta_1-\delta_2) - V_2'V_3'Y_{23}\cos(\theta_{23}+\delta_3-\delta_2) \tag{3b}$$

$$M_3\ddot{\delta}_3 = P_{in_3} - V_3'^2 Y_{33}\cos\theta_{33} - V_1'V_3'Y_{13}\cos(\theta_{13}+\delta_1-\delta_3) - V_2'V_3'Y_{23}\cos(\theta_{23}+\delta_2-\delta_3). \tag{3c}$$

The notation for this and other equations relating to the sample system is given in the Appendix.

In the case when Y_{23} is larger than Y_{12} and Y_{13} and when M_2 and M_3 are smaller than M_1, machines 2 and 3 will behave as a tightly coupled unit. To more readily display this behavior, we rewrite (3) in terms of the variables



Note: All values given in per unit on 100 Mva base. Machine reactances include unit transformer and transient reactance.

Figure 1. Three machine power system.

$$\delta_{c1} = \frac{M_2\delta_2+M_3\delta_3}{M_2+M_3} - \delta_1 \quad , \quad \delta_{23} = \delta_2-\delta_3 \tag{4}$$

as a fourth order system

$$\begin{aligned}\ddot{\delta}_{c1} = {} & \frac{P_{in_2}+P_{in_3}-V_2'^2Y_{22}\cos\theta_{22}-V_3'^2Y_{33}\cos\theta_{33}}{M} - \frac{P_{in_1}-V_1'^2Y_{11}\cos\theta_{11}}{M_1} \\ & - \frac{V_1'V_2'Y_{12}}{M_{12}}\cos(\Psi_{12}-\frac{M_3}{M}\delta_{23}-\delta_{c1}) - \frac{V_1'V_3'Y_{13}}{M_{13}}\cos(\Psi_{13}+\frac{M_2}{M}\delta_{23}-\delta_{c1}) \\ & - \frac{2V_2'V_3'}{M}Y_{23}\cos\theta_{23}\cos\delta_{23}\end{aligned} \tag{5a}$$

$$\begin{aligned}\ddot{\delta}_{23} = {} & \frac{P_{in_2}-V_2'^2Y_{22}\cos\theta_{22}}{M_2} - \frac{P_{in_3}-V_3'^2Y_{33}\cos\theta_{33}}{M_3} - \frac{V_1'V_2'Y_{12}}{M_2}\cos(\theta_{12}-\delta_{c1} \\ & - \frac{M_3}{M}\delta_{23}) + \frac{V_1'V_3'Y_{13}}{M_3}\cos(\theta_{13}-\delta_{c1}+\frac{M_2}{M}\delta_{23}) - \frac{V_2'V_3'Y_{23}}{M_{23}}\cos(\Psi_{23}-\delta_{23})\end{aligned} \tag{5b}$$

where δ_1 is used as the reference.

These equations may be interpreted as follows: Equation (5a) describes the motion of the center of inertia of machines 2 and 3 relative to machine 1. δ_{c1} may be thought of as an average value of δ_2 versus δ_1 and δ_3 versus δ_1. In many cases, the center of inertia of machines 2 and 3 will swing relatively slowly with respect to machine 1. Equation (5b) describes the motion of machine 2 with respect to machine 3. In the case of strong coupling, that is when Y_{23}/M_{23} is larger than Y_{12}/M_{12} and Y_{13}/M_{13}, this motion will be of relatively high frequency. Hence, we see that if

Y_{23}/M_{23} is large, the system (5) fulfills the general requirement of separation of frequencies.

Let us now linearize (5) about the equilibrium point δ^o_{c1}, δ^o_{23}, resulting in the following equations for the perturbations $\Delta\delta_{c1} = \delta_{c1} - \delta^o_{c1}$ and $\Delta\delta_{23} = \delta_{23} - \delta^o_{23}$

$$\Delta\ddot{\delta}_{c1} = -a_{11}\Delta\delta_{c1} + a_{12}\Delta\delta_{23} \tag{6a}$$

$$\Delta\ddot{\delta}_{23} = a_{21}\Delta\delta_{c1} - a_{22}\Delta\delta_{23}. \tag{6b}$$

The frequencies of oscillation of this system will be greatly influenced by a_{11} and a_{22}. From the Appendix, the expressions for a_{11} and a_{22} are

$$a_{11} = \frac{V_1'V_2'Y_{12}}{M_{12}} \sin(\Psi_{12} - \frac{M_3}{M}\delta^o_{23} - \delta^o_{c1}) + \frac{V_1'V_3'Y_{13}}{M_{13}} \sin(\Psi_{13} + \frac{M_2}{M}\delta^o_{23} - \delta^o_{c1}) \tag{7a}$$

$$a_{22} = \frac{V_1'V_2'Y_{12}}{M_2}\frac{M_3}{M} \sin(\theta_{12} - \delta^o_{c1} - \frac{M_3}{M}\delta^o_{23}) + \frac{V_1'V_3'Y_{13}}{M_3}\frac{M_2}{M} \sin(\theta_{13} - \delta^o_{c1} + \frac{M_2}{M}\delta^o_{23})$$
$$+ \frac{V_2'V_3'Y_{23}}{M_{23}} \sin(\Psi_{23} - \delta^o_{23}). \tag{7b}$$

In (7) the role of the quantities Y_{12}/M_{12}, Y_{13}/M_{13}, and Y_{23}/M_{23} in determining the natural frequencies of the system is clearly seen. When Y_{23}/M_{23} is large compared to Y_{12}/M_{12} and Y_{13}/M_{13}, a_{22} is large compared to a_{11}; therefore the response of the system will consist of high and low frequency components.

Since a_{22} takes on a large value when machines 2 and 3 are strongly coupled and since a large value of a_{22} results in high frequency oscillations in the system (6), we let

$$\frac{1}{\mu^2} = a_{22}. \tag{8}$$

Returning to (2) this corresponds to Q_4/μ^2.

We now carry out an analysis of the general system (1) which when written out is

$$\ddot{s}_1 + Q_1 s_1 + \frac{Q_2}{\mu^2} s_2 = 0 \tag{9a}$$

$$\ddot{s}_2 + Q_3 s_1 + \frac{Q_4}{\mu^2} s_2 = 0. \tag{9b}$$

Multiplying (9b) by μ^2 and rearranging yields

$$\mu^2\ddot{s}_2 = -\mu^2 Q_3 s_1 - Q_4 s_2. \tag{10}$$

Following [6] we assume that the variable s_1 is primarily slow and that s_2 can be represented as a sum of its slow and fast parts,

$$s_1 \cong \bar{s}_1 \tag{11a}$$

$$s_2 \cong \bar{s}_2 + \tilde{s}_2 \tag{11b}$$

where the overbar and tilde denote respectively the "slow" part and "fast" part of these variables. An estimate of the error produced by these approximations is given in [4].

Substituting (11) into (10) gives

$$\mu^2\ddot{\bar{s}}_2 + \mu^2\ddot{\tilde{s}}_2 = -\mu^2 Q_3\bar{s}_1 - Q_4\bar{s}_2 - Q_4\tilde{s}_2. \tag{12}$$

Since μ is small and $\bar{s}_2$ is slow, the term $\mu^2\ddot{\bar{s}}_2$ is negligible; that is $\mu^2\ddot{\bar{s}}_2 \cong 0$. Then equating the slow parts in (12) gives

$$0 = -\mu^2 Q_3\bar{s}_1 - Q_4\bar{s}_2 \tag{13}$$

while equating the fast parts results in

$$\mu^2\ddot{\tilde{s}}_2 = -Q_4\tilde{s}_2. \tag{14}$$

Equation (13) represents a "quasi-steady-state expression" which defines the relationship between $\bar{s}_1$ and $\bar{s}_2$, the slowly varying parts of s_1 and s_2. Assuming that Q_4 is invertible, we solve (13) for $\bar{s}_2$ in terms of $\bar{s}_1$

$$\bar{s}_2 = -\mu^2 Q_4^{-1} Q_3\bar{s}_1. \tag{15}$$

The variable $\bar{s}_1$ may be thought of as a low frequency input to the high frequency subsystem and $\bar{s}_2$ is approximately the slow part of the fast subsystem response. This follows the basic idea of the so-called technique of averaging [7]. Of course, this approximation becomes more accurate as the low and high frequency separation increases.

Substituting (15) into (9a) gives the following slow subsystem model

$$\ddot{\bar{s}}_1 + (Q_1 - Q_2Q_4^{-1}Q_3)\bar{s}_1 = 0, \quad \bar{s}_1(t_o) = s_1(t_o), \quad \dot{\bar{s}}_1(t_o) = \dot{s}_1(t_o). \tag{16}$$

Returning to (14), the high frequency subsystem model is

$$\mu^2\ddot{\tilde{s}}_2 = -Q_4\tilde{s}_2 \tag{17}$$

and using (11) and (15), the initial conditions are

$$\tilde{s}_2(t_o) = s_2(t_o) + \mu^2 Q_4^{-1} Q_3 s_1(t_o) \tag{18a}$$

$$\dot{\tilde{s}}_2(t_o) = \dot{s}_2(t_o) + \mu^2 Q_4^{-1} Q_3 \dot{s}_1(t_o). \tag{18b}$$

The fast subsystem can be analyzed in an expanded time scale by introducing the following change of variables in (17)

$$\tau = \frac{t-t_o}{\mu} \tag{19}$$

which gives

$$\frac{d^2\tilde{s}_2}{d\tau^2} = -Q_4\tilde{s}_2. \tag{20}$$

This technique of separating time scales can be extended to nonlinear systems as well [1].

We now use this technique to analyze the system (6). Using the definition of μ^2 from (8) we may write (6) in the form of (9) as follows

$$\Delta\ddot{\delta}_{c1} + a_{11}\Delta\delta_{c1} - \mu^2 \frac{a_{12}}{\mu^2}\Delta\delta_{23} = 0 \tag{21a}$$

$$\Delta\ddot{\delta}_{23} - a_{21}\Delta\delta_{c1} + \frac{1}{\mu^2}\Delta\delta_{23} = 0. \tag{21b}$$

Applying (15), $\Delta\bar{\delta}_{23}$ is given by

$$\Delta\bar{\delta}_{23} = \mu^2 a_{21}\Delta\bar{\delta}_{c1} \tag{22}$$

which when substituted into (21a) yields the following for the low frequency model

$$\Delta\ddot{\bar{\delta}}_{c1} + (a_{11} - \mu^2 a_{12}a_{21})\Delta\bar{\delta}_{c1} = 0. \tag{23}$$

From (17), the high frequency model is

$$\mu^2\Delta\ddot{\tilde{\delta}}_{23} = -\Delta\tilde{\delta}_{23}. \tag{24}$$

We may readily solve (23) and (24) by hand

$$\Delta\bar{\delta}_{c1} = A\cos\omega_1 t + B\sin\omega_1 t \tag{25a}$$

$$\Delta\tilde{\delta}_{23} = C\cos\omega_2 t + D\sin\omega_2 t \tag{25b}$$

where

$$\omega_1 = \sqrt{a_{11} - \mu^2 a_{12}a_{21}} \tag{26a}$$

$$\omega_2 = \frac{1}{\mu}. \tag{26b}$$

The initial conditions are

$$\Delta\bar{\delta}_{c1}(t_o) = \Delta\delta_{c1}(t_o) \tag{27a}$$

$$\Delta\dot{\bar{\delta}}_{c1}(t_o) = \Delta\dot{\delta}_{c1}(t_o) \tag{27b}$$

$$\Delta\tilde{\delta}_{23}(t_o) = \Delta\delta_{23}(t_o) - \mu^2 a_{21}\Delta\delta_{c1}(t_o) \tag{27c}$$

$$\Delta\dot{\tilde{\delta}}_{23}(t_o) = \Delta\dot{\delta}_{23}(t_o) - \mu^2 a_{21}\Delta\dot{\delta}_{c1}(t_o). \tag{27d}$$

The following approximate expressions for $\Delta\delta_{c1}$ and $\Delta\delta_{23}$ may now be written

$$\Delta\delta_{c1}(t) \cong \Delta\bar{\delta}_{c1}(t) \tag{28a}$$

$$\begin{aligned} \Delta\delta_{23}(t) &\cong \Delta\bar{\delta}_{23}(t) + \Delta\tilde{\delta}_{23}(t) \\ &= \mu^2 a_{21}\Delta\bar{\delta}_{c1}(t) + \Delta\tilde{\delta}_{23}(t). \end{aligned} \tag{28b}$$

Finally, we recall that

$$\delta_{c1}(t) = \delta^o_{c1} + \Delta\delta_{c1}(t) \tag{29a}$$

$$\delta_{23}(t) = \delta^o_{23} + \Delta\delta_{23}(t). \tag{29b}$$

Hence, the following approximate solutions for the angles may be written:

$$\delta_{c1}(t) \cong \delta^o_{c1} + \Delta\bar{\delta}_{c1}(t) \tag{30a}$$

$$\delta_{23}(t) \cong \delta^o_{23} + \mu^2 a_{21}\Delta\bar{\delta}_{c1}(t) + \Delta\tilde{\delta}_{23}(t). \tag{30b}$$

For a numerical example, suppose that the system shown in Figure 1 is operating in the steady state and then is disturbed at $t_o = 0$ by the opening of one transmission line from bus 1 to bus 2. Using the numerical values given, the following expressions are obtained

$$\delta_{c1}(t) \cong 32.86^\circ - 18.75^\circ \cos 8.130t \tag{31a}$$

$$\delta_{23}(t) \cong -1.88^\circ + 0.3435 \cos 8.130t - 0.3235 \cos 26.02t. \tag{31b}$$

Note that (31) is given in electrical degrees since this unit is in more common use than electrical radians.

Figure 2 shows a plot of $\delta_{c1}(t)$ as obtained from (5a) and (31a) for a period of 1 second following the opening of the line from bus 1 to bus 2. Figure 3 shows a similar plot for $\delta_{23}(t)$ from (5b) and (31b). We note that for this relatively small disturbance, the agreement is excellent.

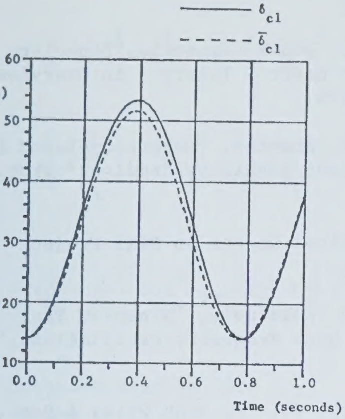


Figure 2. Plot of accurate and approximate δ_{c1}.

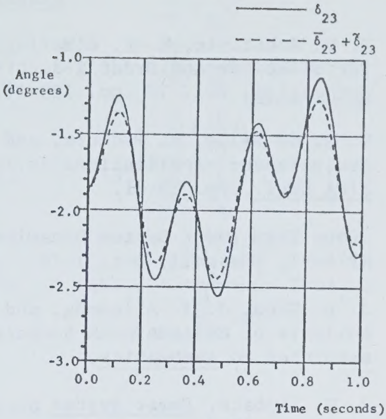


Figure 3. Plot of accurate and approximate δ_{23}.

Examining Figure 3 reveals that the frequency of δ_{23} is about 3 times that of δ_{c1}. The amplitude of δ_{23} is much smaller than the amplitude of δ_{c1}. A strong connection between the machines implies that in the steady state the internal variables will be nearly identical. This suggests that the initial condition of δ_{23} will be small. Hence, we expect the amplitude of the intermachine oscillation to be small. Furthermore, the presence of the slowly varying average value of δ_{23} is readily apparent.

These results may be useful in the on-going study of the coherent behavior of synchronous machines during power system disturbances [2] and may compliment results obtained by dynamic energy balance [8] and long term dynamics techniques [3].

3. CONCLUSIONS

Singular perturbation techniques have been applied to systems which have both low and high frequency behavior. Our procedure consists of decomposing the original system into a slowly varying subsystem and a fast oscillatory subsystem. Using these subsystems, we obtain approximations of the variables of the original system. Besides the computational advantages of dealing with the lower order subsystems, the concept of subsystems contributes to the understanding of structural properties of physical systems. A system of three interconnected synchronous machines has served as an example.

REFERENCES

[1] P. V. Kokotovic, R. E. O'Malley, Jr., and P. Sannuti, "Singular Perturbations and Order Reduction in Control Theory - An Overview," Automatica, Vol. 12, pp. 123-132, 1976.

[2] R. W. de Mello, R. Podmore, and K. N. Stanton, "Coherency-Based Dynamic Equivalents: Applications in Transient Stability Studies," Proc. 1975 PICA Conf., pp. 23-31.

[3] "Long Term Power System Dynamics," Final Report on EPRI Project RP764-1, Phase II, Oct. 1976.

[4] J. H. Chow, J. J. Allemong, and P. V. Kokotovic, "Singular Perturbation Analysis of Systems with Sustaining High Frequency Oscillations," submitted to Automatica.

[5] E. W. Kimbark, Power System Stability, Vol. 1, John Wiley & Sons, New York, 1948.

[6] J. H. Chow and P. V. Kokotovic, "A Decomposition of Near-Optimum Regulators for Systems with Slow and Fast Modes," IEEE Trans. on Automatic Control, Vol. AC-21, pp. 701-705, Oct. 1976.

[7] N. N. Bogoluibov and Yu. A. Mitropolskii, Asymptotic Methods in the Theory of Nonlinear Oscillations, Gordon and Breach, New York, 1961.

[8] K. N. Stanton, "Power System Dynamic Simulation Using Models with Reduced Dimensionality," Proc. JACC, 1972, pp. 415-419.

APPENDIX

Notation used in (3)

δ_i: rotor angle of machine i in electrical radians.

M_i: inertia constant of machine i in $\sec^2$/elect. rad.

P_{in_i}: input power to machine i in per unit.

V_i': voltage behing transient reactance of machine i in per unit.

Y_{ij}: per unit magnitude of the ijth element of the reduced network admittance matrix.

θ_{ij}: angle in radians of the ijth element of the reduced network admittance matrix.

Notation used in (5)

$M = M_2 + M_3$

$$(\frac{1}{M_{12}})^2 = (\frac{1}{M} - \frac{1}{M_1})^2 \cos^2\theta_{12} + (\frac{1}{M} + \frac{1}{M_1})^2 \sin^2\theta_{12}, \quad \tan\Psi_{12} = \frac{(\frac{1}{M} + \frac{1}{M_1})}{(\frac{1}{M} - \frac{1}{M_1})} \tan\theta_{12}$$

$$(\frac{1}{M_{13}})^2 = (\frac{1}{M} - \frac{1}{M_1})^2 \cos^2\theta_{13} + (\frac{1}{M} + \frac{1}{M_1})^2 \sin^2\theta_{13}, \quad \tan\Psi_{13} = \frac{(\frac{1}{M} + \frac{1}{M_1})}{(\frac{1}{M} - \frac{1}{M_1})} \tan\theta_{13}$$

$$(\frac{1}{M_{23}})^2 = (\frac{1}{M_2} - \frac{1}{M_3})^2 \cos^2\theta_{23} + (\frac{1}{M_2} + \frac{1}{M_3})^2 \sin^2\theta_{23}, \quad \tan\Psi_{23} = \frac{(\frac{1}{M_2} + \frac{1}{M_3})}{(\frac{1}{M_2} - \frac{1}{M_3})} \tan\theta_{23}.$$

Notation used in (6)

$$a_{11} = \frac{V_1'V_2'Y_{12}}{M_{12}} \sin(\Psi_{12} - \frac{M_3}{M}\delta^o_{23} - \delta^o_{c1}) + \frac{V_1'V_3'Y_{13}}{M_{13}} \sin(\Psi_{13} + \frac{M_2}{M}\delta^o_{23} - \delta^o_{c1})$$

$$a_{12} = -\left[\frac{V_1'V_2'Y_{12}}{M_{12}} \frac{M_3}{M} \sin(\Psi_{12} - \frac{M_3}{M}\delta^o_{23} - \delta^o_{c1}) - \frac{V_1'V_3'Y_{13}}{M_{13}} \frac{M_2}{M} \sin(\Psi_{13} + \right.$$
$$\left. + \frac{M_2}{M}\delta^o_{23} - \delta^o_{c1}) - \frac{2V_2'V_3'}{M} Y_{23} \cos\theta_{23} \sin\delta^o_{23}\right]$$

$$a_{21} = -\left[\frac{V_1'V_2'Y_{12}}{M_2} \sin(\theta_{12} - \delta^o_{c1} - \frac{M_3}{M}\delta^o_{23}) - \frac{V_1'V_3'Y_{12}}{M_2} \sin(\theta_{13} - \delta^o_{c1} + \frac{M_2}{M}\delta^o_{23})\right]$$

$$a_{22} = \frac{V_1'V_2'Y_{12}}{M_2} \frac{M_3}{M} \sin(\theta_{12} - \delta^o_{c1} - \frac{M_3}{M}\delta^o_{23}) + \frac{V_1'V_3'Y_{13}}{M_3} \frac{M_2}{M} \sin(\theta_{13} - \delta^o_{c1} + \frac{M_2}{M}\delta^o_{23})$$
$$+ \frac{V_2'V_3'Y_{23}}{M_{23}} \sin(\Psi_{23} - \delta^o_{23}).$$

CLOSED LOOP COORDINATED CONTROL OF LARGE POWER PLANTS*

N. GOPALSAMI and C.K. SANATHANAN
Department of Information Engineering
University of Illinois at Chicago Circle
Chicago, Illinois 60680

ABSTRACT

A practically implementable feedback control scheme for a large-scale system composed of a number of interconnected subsystems, as exemplified by a power plant, is proposed in this paper. The design proceeds in two steps. First, the local controllers for the subsystems are independently designed ignoring the interconnections. Then a global feedback control on a higher hierarchical level is designed from the interconnection variables. The resulting parameter optimization problem on the higher level is solved by a decentralized computational structure. It is claimed that most of the deterioration of the local optima caused by the interactions may be compensated in this manner. Even so, the restricted structure of the feedback scheme makes the resulting control suboptimal. It is worthwhile to determine the closeness of the suboptimal index to the optimal one without computing the optimal solution. For this, it is shown that the optimal performance index is bounded between the suboptimal index and a lower bound computed from the dual suboptimal system. The closeness between the bounds may be used to judge the merit of the suboptimal design.

I. INTRODUCTION

Due to the high dimensionality and the inevitable presence of nonclassical information structure, the control and optimization of large power plants is generally done in the framework of decomposition and coordination. Much work has been done in recent years utilizing the principles of hierarchical multilevel thoery [1-3]. In these methods, the large dynamic optimization problem is broken up into a number of smaller subproblems by fixing certain interaction trajectories (called the coordination variables). This enables one to solve these subproblems independently. On a higher level, the coordination variables are successively modified in order to force the independent subproblem solutions to the optimal solution of the overall problem.

The inherent drawback with these methods is that they generally yield open loop control at the higher level even though the local controllers can be synthesized in the feedback form. Singh et al [3] have modified the open-loop part of the controls by linearly relating these controls to the states of the overall system where by the overall control is presented in feedback form. However, this requires an enormous amount of computation over and above those for the hierarchical open-loop solution.

Alternately, Siljak et al [4-5] have developed a multilevel feedback scheme by treating the interconnections as perturbations acting on the individual subsystems. The principal advantage of this approach is the reliability of the system performance under structural perturbation and the noniterative computational scheme for the higher level (or global) control. However, this method does not aim at achieving the overall optimum of the composite system. Also, the design of the global controller is done without any consideration of its effect in the overall cost function.

---

*Work performed under NSF Grant No. ENG76-09930.

In this paper, a practically implementable multilevel feedback control is described for decentralized optimization of a large-scale system composed of a number of interconnected subsystems as exemplified by a large power plant. The local controllers are designed for the individual subsystems in a completely decentralized environment ignoring the interconnections. A global controller on a higher level is then used to minimize the degradation due to interconnections on the performance of the overall system. Since any degradation of the local optimums is due to interconnections, it is proposed that the global controls may be synthesized by feeding back the interconnection variables. This not only simplifies the computational effort but more importantly results in a closed loop control as well. The parameter optimization problem is solved in a decentralized computational structure employing the interaction prediction principle [6].

When the optimal solution is highly cumbersome to compute as in the case of a large power plant, it is impossible to judge beforehand how close the suboptimal control is to the optimal one. For this, Canales [8] has given a lower bound on the optimal performance index. The method, though elegant theoretically, becomes extremely complicated for large systems. Using the results of McClamroch [7], it is shown in this paper that the optimal performance index is bounded between the suboptimal index and a lower bound. The latter is obtained from the dual suboptimal system whose control is given in terms of the suboptimal solution of the primal problem. The closeness between the bounds may be used to judge the merit of the suboptimal design.

II. A MULTILEVEL FEEDBACK SCHEME

Consider a large-scale system composed of N interconnected subsystems described as follows:

$$\dot{x}_i = A_i x_i + B_i u_i + C_i z_i \qquad i = 1, 2, \ldots, N \tag{1}$$

$$z_i = \sum_{j=1}^{N} L_{ij} x_j \qquad i = 1, 2, \ldots, N \tag{2}$$

where x_i is an n_i dimensional state vector of the ith subsystem, u_i is the m_i dimensional control vector, and z_i is the q_i dimensional vector of interconnections from other subsystems. A_i, B_i, C_i, and L_{ij} are appropriately dimensioned constant matrices. It is assumed that the pairs (A_i, B_i) are completely controllable and also the composite system is assumed to be completely controllable. The optimization problem considered here is to minimize

$$J = \sum_{i=1}^{N} \int_{t_o}^{t_f} \frac{1}{2}\left(\left\| x_i \right\|_{Q_i}^2 + \left\| u_i \right\|_{R_i}^2\right) dt \tag{3}$$

where $\|s\|_L^2 = s^T L\, s$, Q_i and R_i are symmetric positive definite matrices, subject to the constraints (1) and (2).

When the subsystems are decoupled i.e., when $z_i = 0$ for $i = 1, 2, \ldots, N$, the problem of minimization of J in (3) reduces to N independent linear quadratic problems with classical information structure. The local optimal controls $u_i^*(t)$ that minimize

$$J_i = \int_{t_o}^{t_f} \frac{1}{2}\left(\left\| x_i \right\|_{Q_i}^2 + \left\| u_i \right\|_{R_i}^2\right) dt \qquad i = 1, 2, \ldots, N \tag{4}$$

subject to the constraints

$$\dot{x}_i = A_i x_i + B_i u_i \qquad i = 1, 2, \ldots, N \tag{5}$$

are given by

$$u_i^*(t) = -K_i\, x_i(t) = -R_i^{-1}\, B_i^T\, P_i\, x_i(t) \tag{6}$$

P_i being the symmetric positive definite solution of the Riccati equation

$$-\dot{P}_i = P_i\, A_i + A_i^T\, P_i - P_i\, B_i\, R_i^{-1}\, B_i^T\, P_i + Q_i \tag{7}$$

$$\text{with } P_i(t_f) = 0$$

These controls result in the minimum cost

$$J_i^* = \tfrac{1}{2}\, x_i^T(t_o)\, P_i\, x_i(t_o) \qquad i = 1, 2, \ldots, N \tag{8}$$

The composite system with the application of these local controls becomes

$$\dot{x}_i = (A_i - B_i\, K_i)x_i(t) + C_i\, z_i(t) \qquad i = 1, 2, \ldots, N \tag{9}$$

The resulting performance index generally deviates from the decoupled optimums due to the presence of interactions. Hence, certain corrective controls (called global controls) are introduced in addition to the decentralized locally optimal controls $u_i^*(t)$. It is logical that these global controls would have strong dependence on the interaction variables z_i, $i = 1, 2, \ldots, N$.

The input $u_i(t)$ to each subsystem is expressed as:

$$u_i(t) = u_i^*(t) + u_i^g(t) \tag{10}$$

where $u^*(t) = -K_i\, x_i(t)$ is the locally optimal control and $u_i^g(t)$ is the global control assumed to be of the form

$$u_i^g(t) = -M_i\, z_i(t) \qquad i = 1, 2, \ldots, N \tag{11}$$

where M_i $(i = 1, 2, \ldots, N)$ are $m_i \times q_i$ constant (parameter) matrices to be chosen so as to minimize J in (3). More precisely, the following parameter optimization problem is solved in the higher level. Choose M_i, $i = 1, \ldots, N$ to minimize

$$J = \sum_{i=1}^{N} \int_{t_o}^{t_f} \frac{1}{2} \left\| x_i \right\|_{Q_i}^2 + \left\| -K_i\, x_i - M_i\, z_i \right\|_{R_i}^2 \, dt \tag{12}$$

subject to

$$\dot{x}_i = (A_i - B_i\, K_i)x_i(t) + C_i\, z_i(t) + B_i\, u_i^g(t) \tag{13}$$

$$z_i(t) = \sum_{j=1}^{N} L_{ij}\, x_j(t) \tag{14}$$

$$u_i^g = -M_i\, z_i(t) \tag{15}$$

The above problem may be solved by using the interaction prediction principle [6].

Define the dual function

$$\phi(\lambda) = \min_{\substack{x_i, M_i \\ i=1,\ldots,N}} \{L_i(x_i, M_i, \lambda, z_i)\} \tag{16}$$

where

$$L = \sum_{i=1}^{N} L_i = \sum_{i=1}^{N} \int_{t_o}^{t_f} \Big[\tfrac{1}{2} \Big(\left\| x_i \right\|_{Q_i}^{2} + \left\| -K_i x_i - M_i z_i \right\|_{R_i}^{2} \Big) + p_i^T \{ -\dot{x}_i + A_i x_i - B_i K_i x_i + (C_i - B_i M_i) z_i \} + \lambda_i^T \big(z_i - \sum_{j=1}^{N} L_{ij} x_j \big) \Big] dt \tag{17}$$

Then by the theorem of strong Lagrange duality

$$\max_{\lambda} \phi(\lambda) = \min_{\substack{M_i \\ i=1,\dots,N}} J \tag{18}$$

Rewriting the Lagrangian of (17),

$$L = \sum_{i=1}^{N} L_i = \sum_{i=1}^{N} \int_{t_o}^{t_f} \Big[\tfrac{1}{2} \Big(\left\| x_i \right\|_{Q_i}^{2} + \left\| -K_i x_i - M_i z_i \right\|_{R_i}^{2} \Big) + p_i^T \{ -\dot{x}_i + A_i x_i - B_i K_i x_i + (C_i - B_i M_i) z_i \} + \lambda_i^T z_i - \sum_{j=1}^{N} \lambda_j^T L_{ji} x_i \Big] dt \tag{19}$$

This implies that for any given z_i, λ trajectories, L is additively separable and the minimum of $\phi(\lambda)$ with respect to x_i, M_i can be obtained by N independent minimizations. This leaves the problem of improving z_i, λ such that the maximum of $\phi(\lambda)$ is achieved.

A necessary condition for maximizing $\phi(\lambda)$ is that

$$\frac{\partial L}{\partial z_i} = 0 \quad \text{and} \quad \frac{\partial L}{\partial \lambda} = 0 \tag{20}$$

which gives the following iterative scheme.

$$\begin{bmatrix} \lambda_i \\ \\ z_i \end{bmatrix}^{k+1} = \begin{bmatrix} -(C_i - B_i M_i)^T p_i - M_i^T R_i (K_i x_i + M_i z_i) \\ \\ \sum_{j=1}^{N} L_{ij} x_j \end{bmatrix}^{k} \tag{21}$$

The convergence of this algorithm has been proved by Takahara [6]. The important point to note is that very little work is involved in the computation of (21) and the necessary conditions for the minimization of L_i in (19) lead to a linear Two Point Boundary Value Problem (TPBVP) rather than the usual nonlinear TPBVP. Also, the converged solution (or even the intermediate solution) results in a feasible closed loop control.

The restricted structure of the overall control (incomplete state feedback), however, can yield only a suboptimal solution. Since it is conjectured that the overall optimum of the large system is impossible to be computed, a method of computing its lower bound is given in the next section.

III. BOUNDS ON THE OPTIMAL PERFORMANCE INDEX

The results in this section are given for the infinite time linear regulator problem. It is assumed at this point that a suboptimal control law is available which results in a performance index, say J_u and it is of interest to determine a lower bound, J_ℓ on the optimal index J^* such that

$$J_\ell \leq J^* \leq J_u \tag{22}$$

Consider the original system (1) and (2) written in the form

$$\dot{x} = \bar{A}\,x + \bar{B}\,u \qquad x(t_o) = x_0 \tag{23}$$

$$y = x \tag{24}$$

with the performance index

$$J = \frac{1}{2}\int_{t_o}^{\infty} \left(\|y\|_Q^2 + \|u\|_R^2 \right) dt \tag{25}$$

where $\bar{A}$, $\bar{B}$, Q and R are the aggregated composite matrices and y is the output vector (assumed for simplicity to be the state vector itself).

The dual problem of (23)-(25) is given by

$$\dot{x} = -\bar{A}^T\,x + y(t) \qquad x(t_o) = x_o \tag{26}$$

$$u(t) = \bar{B}^T\,x(t) \tag{27}$$

$$J = \frac{1}{2}\int_{t_o}^{\infty} \left(\|u\|_{R^{-1}}^2 + \|y\|_{Q^{-1}}^2 \right) dt \tag{28}$$

where y(t) is to be chosen such that J is minimized.

The main result of this section is stated in the following theorem.

Theorem. Assume that the composite system is completely controllable and observable. Let $u = -K_o\,x(t)$ be a suboptimal stable control law for the primal problem given in (23)-(25) and let the symmetric positive definite cost matrix associated with it be denoted by P_1. Then for the dual problem, the control law $y = -Q\,G\,x(t)$ where $G = [I + \frac{1}{2}Q^{-1}(K_o - R^{-1}\bar{B}^T P_1)^T R(K_o - R^{-1}\bar{B}^T P_1)]P_1^{-1}$ results in a stable closed loop system and if the associated cost matrix is N_1, then

$$N_1^{-1} \leq P^* \leq P_1$$

where P^* is the cost matrix associated with the optimal control law of the primal problem.

Proof: It can be shown [10] that for any K_o for which $(\bar{A} - \bar{B}\,K_o)$ has eigenvalues with negative real parts, the cost functional J becomes

$$J = \tfrac{1}{2}\,x(t_o)^T\,P_1\,x(t_o) \tag{29}$$

where P_1 is the unique $n \times n$ symmetric positive definite matrix (called the cost matrix) satisfying

$$(\bar{A} - \bar{B}\,K_o)^T\,P_1 + P_1(\bar{A} - \bar{B}\,K_o) + K_o^T\,R\,K_o + Q = 0 \tag{30}$$

Rearranging (30),

$$-P_1\,\bar{A} - \bar{A}^T\,P_1 + P\,\bar{B}\,K_o + K_o^T\,\bar{B}^T\,P_1 - Q - K_o^T\,R\,K_o = 0 \tag{31}$$

Adding to both sides of the above equation,

$$-2Q = -Q\,P_1^{-1}\,P_1 - P_1\,P_1^{-1}\,Q,$$

(31) becomes

$$P_1(-\bar{A} - P_1^{-1}\,Q) + (-\bar{A}^T - Q\,P_1^{-1})P_1 = -Q + K_o^T R\,K_o - P_1\,\bar{B}\,K_o - K_o^T\,\bar{B}^T\,P_1 \tag{32}$$

Let $K_1 = R^{-1} \bar{B}^T P_1 = K_o - D$

Then (32) becomes,

$$P_1(-\bar{A} - P_1^{-1} Q) + (-\bar{A}^T - Q P_1^{-1})P_1 = -Q - K_1^T R K_1 + D^T R D \tag{33}$$

Rearranging (33),

$$P_1^{-1}(-\bar{A}^T - Q P_1^{-1} - \tfrac{1}{2} D^T R D P_1^{-1}) + (-\bar{A}^T - Q P_1^{-1} - \tfrac{1}{2} D^T R D P_1^{-1})^T P_1^{-1} = -P_1^{-1}(Q + K_1^T R K_1)P_1^{-1} \tag{34}$$

Since P_1^{-1} is positive definite and the right hand side of (34) is negative definite, Lyapunov stability theory guarantees that $[-\bar{A}^T - (Q + \frac{1}{2} D^T R D)P_1^{-1}]$ has eigenvalues with negative real parts [9]. Hence,

$$y = -Q[I + \tfrac{1}{2} Q^{-1} D^T R D]P_1^{-1} x(t) = -L_o x(t) \tag{35}$$

is a stable control law for the dual system and the corresponding cost is given by

$$J = \tfrac{1}{2} x(t_o)^T N_1 x(t_o)$$

where N_1 is the symmetric positive definite matrix which satisfies

$$N_1[-\bar{A}^T - Q P_1^{-1} - \tfrac{1}{2} D^T R D P_1^{-1}] + [-\bar{A}^T - Q P_1^{-1} - \tfrac{1}{2} D^T R D P_1^{-1}]^T N_1 + L_o^T Q L_o + \bar{B} R^{-1} \bar{B}^T = 0 \tag{36}$$

Now, using the result obtained by McClamroch [7], it can be shown readily that:

$$N_1^{-1} \leq P^* \leq P_1 \tag{37}$$

This proves the theorem.

It is known that if N^* is the optimal cost matrix for the dual problem, then

$$N^* = (P^*)^{-1} \tag{38}$$

and also that the optimal control law for the dual problem is given by

$$y^* = -Q(P^*)^{-1} x(t) \tag{39}$$

Considering the control law given in (35), it is seen that when P_1 is close to P^* in the sense that $x(t_o)(P_1 - P^*) x(t_o)$ is small, $\frac{1}{2} D^T R D$ is negligible and so,

$$y \simeq -Q P_1^{-1} x(t)$$

Hence this would result in a cost matrix N_1 close to N^* $(N_1 \geq N^*)$ and N_1^{-1} would be close to P^* $(N_1^{-1} \leq P^*)$ as seen by the relation (38).

As the suboptimal performance value approaches the optimum, the bounds on the optimum value become closer. The degree of suboptimality is thus judged from the closeness between the upper and lower bounds.

Only the infinite time regulator problem has been considered. This is because the results are rather elegant for this case. Since the determination of a lower bound is equivalent to determination of a suboptimal control for the dual problem, the results can in theory be extended to the finite time case. The important consideration is that the dual control law given in (35) should be suitably modified to result in a stable dual suboptimal system.

IV. EXAMPLE

A sixth order system, with three interconnected subsystems each of second order, is considered first for illustration.

System:

$$\dot{x}_1 = x_2 + c_1 z_1$$

$$\dot{x}_2 = -2x_1 - x_2 + u_1 \quad : \quad z_1 = x_4 + x_5$$

$$\dot{x}_3 = x_4$$

$$\dot{x}_4 = -x_3 - 5x_4 + u_2 + c_2 z_2 \quad : \quad z_2 = x_1$$

$$\dot{x}_5 = x_6$$

$$\dot{x}_6 = -x_5 - 10x_6 + u_3 + c_3 z_3 \quad : \quad z_3 = x_1 + x_3$$

Performance indices:

$$J_1 = \frac{1}{2}\int_0^{t_f} (x_1^2 + x_2^2 + u_1^2)dt$$

$$J_2 = \frac{1}{2}\int_0^{t_f} (x_3^2 + x_4^2 + u_2^2)dt$$

$$J_3 = \frac{1}{2}\int_0^{t_f} (x_5^2 + x_6^2 + u_3^2)dt$$

$$J = J_1 + J_2 + J_3$$

Results:

A. Decoupled ($z_i = 0$, $i = 1, 2, 3$)
[$t_f = 10$ sec]
$J_1 = 1.24732$ $J_2 = 1.63261$ $J_3 = 2.40532$

B. Forced coordination (connecting the optimized decoupled subsystems)
[$c_1 = c_2 = c_3 = 1$]
$J_1 = 15.1924$ $J_2 = 4.80472$ $J_3 = 9.09685$

C. Multilevel coordination
$J_1 = 6.92023$ $J_2 = 3.40237$ $J_3 = 5.72080$ $J = 16.04340$

D. Optimal coordination
$J_1 = 6.78586$ $J_2 = 3.6550$ $J_3 = 5.29147$ $J = 15.73233$

E. Infinite time problem
$J^* = 16.0126$
For the control law of case C,
$15.2077 \le J^* \le 17.1113$

The proposed control scheme is applied to a 200MWe reheat drum-type coal fired power plant shown in figure 1. The model of the boiler-turbine unit is developed in detail in [11]. The generator is assumed to be connected to an infinite bus with appropriate voltage regulation. The power plant is required to generate electrical power to meet typically a saturating ramp type of power demand.

The results of the control and optimization study on the power plant are not complete. However, some preliminary results of the CSMP simulation show the effectiveness of the proposed scheme. Each subsystem (the boiler-turbine and the generator) has its own local controllers to keep the various quantities such as turbine speed, steam termperature, etc. near their reference values. A global controller is used to modify the local controls of the throttle valve opening and the coal feeder stroke. For this the error between the power demand and the power generated is used as the actuating signal. Figures 2 and 3 show the superiority of the coordinated control over the conventional boiler following mode of control. From figure 3, it is seen that the oscillations of the coal feeder stroke are damped considerably. This is an important consideration in terms of the thermal stresses and fatigue to the boiler components.

V. CONCLUSION

Excessive data processing requirements and complexity in implementation make conventional optimal control impractical in the case of a large power plant. The closed loop multilevel control scheme, developed in this paper, does not in principle require the measurement of all state variables and is easily implementable. The local controllers for the individual subsystems are designed ignoring the interconnections. The higher level controller is designed to minimize the degradation, due to interconnections, of the overall optimum using a decentralized computational procedure. However, the resulting control is suboptimal. A method of evaluating the closeness of the suboptimal index to the optimal one is provided without solving for the optimal value. In many cases, if the suboptimal value is too far from the optimum, it can be brought close by using a more sophisticated feedback scheme. However, the improvement in performance must be weighed against the increased complexity and reliability of implementation.

REFERENCES

1. M.S. Mahmoud, "Multilevel systems control and applications: a survey," IEEE Trans. Systems, Man, and Cybern., Vol. SMC-7, pp. 125-143, 1977.
2. J.D. Pearson, "Dynamic decomposition techniques," in Optimization Methods for Large-Scale Systems, D.A. Wismer, Ed. New York: McGraw-Hill, 1970.
3. M.G. Singh, M.F. Hassan, and A. Titli, "Multilevel feedback control for interconnected dynamical systems using the predication principle," IEEE Trans. Systems, Man, and Cybern., Vol. SMC-6, pp. 233-239, 1976.
4. D.D. Siljak and M.K. Sundareshan, "A multilevel optimization of large-scale dynamic systems," IEEE Trans. Automat. Control, Vol. AC-21, pp. 79-84, 1976.
5. M.K. Sundareshan, "Generation of multilevel control and estimation schemes for large-scale systems: a perturbational approach," IEEE Trans. Systems, Man and Cybern., Vol. SMC-7, pp. 144-152, 1977.
6. Y. Takahara, "A multilevel structure for a class of dynamical optimization problems," M.S. thesis, Case Western Reserve University, Cleveland, OH, 1965.
7. N.H. McClamroch, "Duality and bounds for the matrix Riccati equation," J. Math. Anal. and Appl., Vol. 25, pp. 622-627, 1969.
8. R. Canales, "A lower bound on the performance of optimal regulators," IEEE Trans. Autom. Control, Vol. AC-15, pp. 409-415, 1970.
9. D.L. Kleinman, "An easy way to stabalize a linear constant system," IEEE Trans. Autom. Control, Vol. AC-15, p. 692, 1970.
10. B.D.O. Anderson and J.B. Moore, Linear Optimal Control, Prentice-Hall, 1971.
11. J.P. McDonald et al., "Nonlinear model of a reheat Boiler-Turbine-Generator System," Research Division Report No. E-196, Philadelphia Electric Company, September 1970.

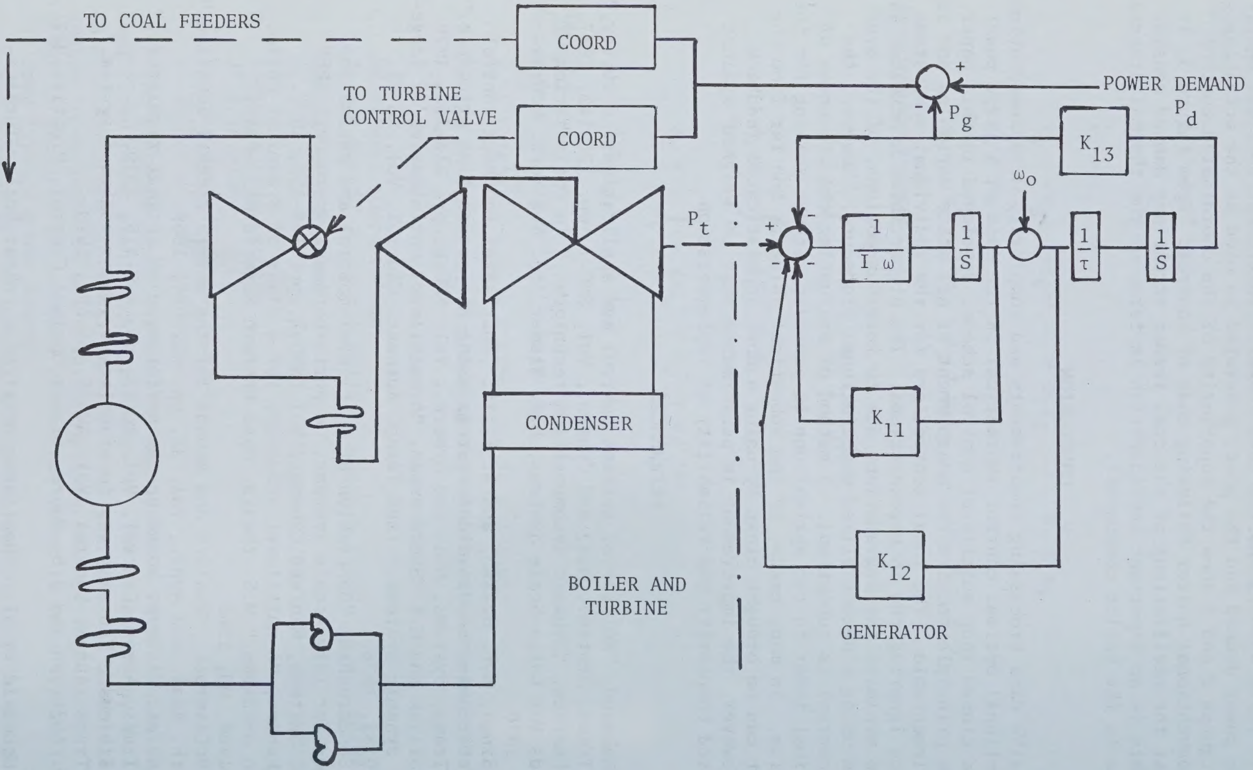

TO COAL FEEDERS
COORD
TO TURBINE
CONTROL VALVE
COORD
+
-
POWER DEMAND
P_d
P_g
K_{13}
ω_o
P_t
+
-
$\frac{1}{I\ \omega}$
$\frac{1}{S}$
$\frac{1}{\tau}$
$\frac{1}{S}$
K_{11}
K_{12}
CONDENSER
BOILER AND
TURBINE
GENERATOR

FIGURE 1. PLANT SCHEMATIC FOR COORDINATED CONTROL

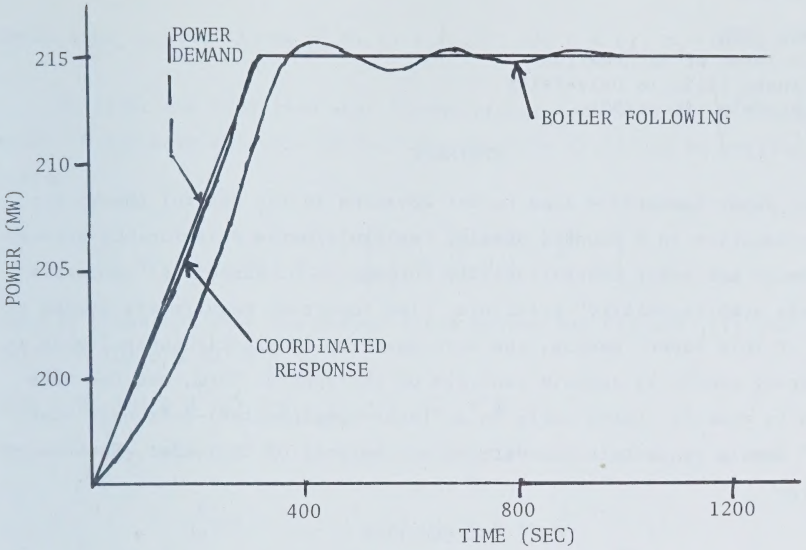


FIGURE 2. POWER Vs. TIME

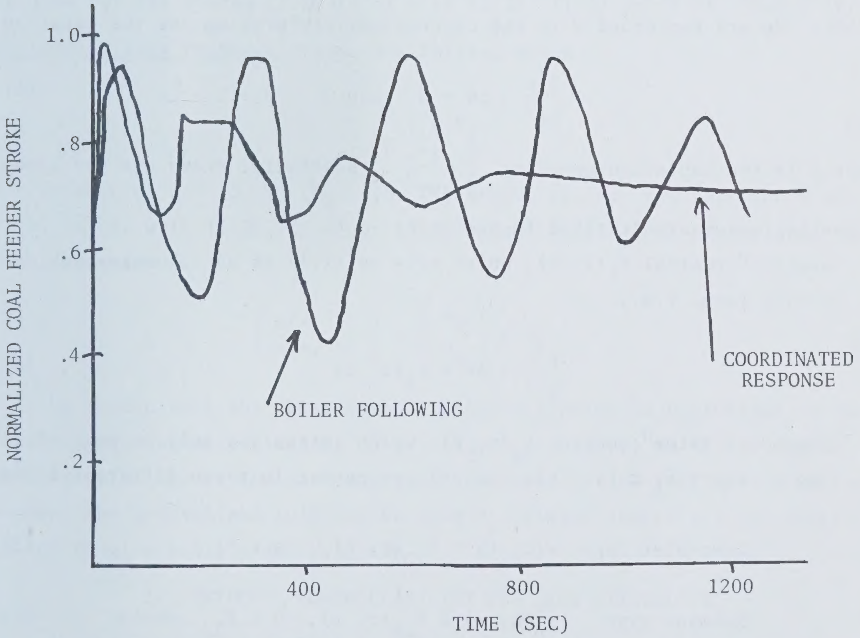


FIGURE 3. COAL FEEDER STOKE Vs. TIME

ENERGY DECAY ESTIMATES AND CONTROL THEORY FOR THE WAVE EQUATION IN A BOUNDED DOMAIN

GOONG CHEN
Department of Mathematics
Southern Illinois University
Carbondale, IL 62901

ABSTRACT

This paper summarizes some recent advances in the control theory for the wave equation in a bounded domain. We explain the relationship between energy decay and exact controllability through D. L. Russell's "controllability via stabilizability" principle. Two important results are stated at the end of this paper, namely, the wave equation is exactly controllable in an arbitrary domain by certain controls of the spatial form, and the wave equation is exactly controllable in a "(star-complemented)-(strongly star-shaped)" domain by certain boundary value controls of Dirichlet, Neumann or Robin type.

1. INTRODUCTION

In this paper, we summarize some recent advances in the control theory for the wave equation in a bounded domain by the energy method.

Let Ω be a bounded open set in $\mathbb{R}^N$ with piecewise smooth boundary Γ ($= \partial\Omega$). We are concerned with the controllability problem for the equation

$$\frac{\partial^2 w}{\partial t^2} - \Delta w = 0 \quad \text{on } \Omega \tag{1}$$

where Δ is the Laplacian operator $\sum_1^N \frac{\partial^2}{\partial x_i^2}$. Basically, there are two types of control which are feasible in practice:

i) "spatial" control $f_1(x, t)$, which acts on (1.1) as an inhomogeneous forcing term, i.e.,

$$\frac{\partial^2 w}{\partial t^2} - \Delta w = f_1(x, t) \tag{2}$$

ii) "boundary value" control $f_2(x, t)$, which is applied only on part of the boundary $\Gamma_1 \subset \Gamma$. This control may appear in three different forms:

$$\text{Dirichlet type} \quad w(\sigma, t) \equiv f_{2D}(\sigma, t)\ , \quad \sigma \in \Gamma_1 \tag{3}$$

$$\text{Neumann type} \quad \frac{\partial w}{\partial n}(\sigma, t) \equiv f_{2N}(\sigma, t), \quad \sigma \in \Gamma_1, \quad \text{where} \tag{4}$$

n is the unit outward normal which

exists almost everywhere on Γ.

Robin type $\alpha_1 w(\sigma, t) + \alpha_2 \frac{\partial w}{\partial n}(\sigma, t) \equiv f_{2R}(\sigma, t), \sigma \in \Gamma_1, \alpha_1 > 0, \alpha_2 > 0$ (5)

We first ask ourselves what is the proper formulation for the state space of the wave equation (1.1). The equation (1.1) can be written as a system

$$\frac{d}{dt}\begin{bmatrix} w \\ v \end{bmatrix} = \begin{bmatrix} 0 & I \\ \Delta & 0 \end{bmatrix}\begin{bmatrix} w \\ v \end{bmatrix} \equiv A\begin{bmatrix} w \\ v \end{bmatrix} \tag{6}$$

Thus it turns out that the proper state spaces for (i) and (ii) are

(i)' $H_0^1(\Omega) \oplus H^0(\Omega) \equiv \mathcal{H}_1$

(ii)' $H_{\Gamma_0}^1(\Omega) \oplus H^0(\Omega) \equiv \mathcal{H}_1, \Gamma_0 = \Gamma - \Gamma_1 \neq \emptyset$

respectively, where $H^k(\Omega)$ denotes the Sobolev space of order k and

$$H_0^k(\Omega) = \{u \in H^k(\Omega) \mid u|_\Gamma = 0 \text{ in } H^{k-1/2}(\Gamma)\}$$

$$H_{\Gamma_0}^k(\Omega) = \{u \in H^k(\Omega) \mid u|_{\Gamma_0} = 0 \text{ in } H^{k-1/2}(\Gamma)\}$$

We pose for the system (1.2) ((1.1) with (1.3), (1.4) or (1.5), resp.) the

Controllability Problem: Given any initial state

$$\begin{bmatrix} w_0 \\ v_0 \end{bmatrix} \in \mathcal{H}_1 \tag{7}$$

determine f on $\Omega \times [0, T]$ ($\Gamma_1 \times [0, T]$, resp.) so that the solution w of (1.2) ((1.1) with (1.3), (1.4) or (1.5) resp.) satisfies a preassigned terminal state

$$\begin{bmatrix} w(\cdot, T) \\ v(\cdot, T) \end{bmatrix} = \begin{bmatrix} w_T \\ v_T \end{bmatrix} \in \mathcal{H}_1. \tag{8}$$

We remark that the above controllability problem is equivalent to the "null controllability problem," i.e., $(w_T, v_T) = (0, 0)$ in (1.8), due to the reversibility of the wave equation. We also notice that the "solution" w means the generalized solution because $\mathcal{H}_1$-initial states are not smooth enough to give a classical solution.

2. METHOD: CONTROLLABILITY VIA STABILIZABILITY

A controllability problem in infinite dimensions can be solved by various methods, for example, the abstract moment problem approach ([7], [12]) or the extension of the finite rank conditions ([15]). But for the wave equation, the most efficient method to obtain exact controllability is D. L. Russell's "controllability via stabilizability" principle. For a finite dimensional control system

$$\frac{d}{dt} x = Ax + Bf , \quad x \in \mathbb{R}^N, \quad f \in \mathbb{R}^M \tag{9}$$

A, B are constant N × N, N × M matrices,

his principle can be stated as follows:

Theorem 2.1. (Russell [11])

Let the system (9) be both (+) and (-) stabilizable, namely, there exist two M × N constant matrices K^+ and K^- such that

$$A^+ \equiv A + BK^+ \tag{10}$$

$$A^- \equiv A + BK^- \tag{11}$$

have, respectively, only eigenvalues with negative and positive real parts. Then the system (9) is controllable.

We refer the readers to [11] for a proof. The main idea is to use feedback signals (cf. [11]) from auxiliary systems

$$f^+(t) = K^+x^+(t) \tag{12}$$

$$f^-(t) = K^-x^-(t). \tag{13}$$

The control f is obtained by "blending" the above signals, i.e.,

$$f(t) \equiv f^+(t) + f^-(t). \tag{14}$$

In the proof, it is essential that we have

$$\|e^{A^+t}\| \leq K_1 e^{-\alpha t} \tag{15}$$

$$\|e^{-A^-t}\| \leq K_2 e^{-\alpha t} \tag{16}$$

for some K_1, K_2, $\alpha > 0$, in order to guarantee the invertibility of a certain transformation when t large.

Theorem 2.1 is readily extended to an infinite-dimensional system after a suitable modification, e.g., by replacing e^{A^+t} and e^{-A^-t} in (15) and (16) by suitable semigroups of operators on H_1 and like things. It is most important to observe that we must retain a uniform decay rate for this kind of semigroups S(t), i.e., we require

$$\|S(t)\|_{L(H_1, H_1)} \leq f(t) \tag{17}$$

where $\lim_{t\to\infty} f(t) = 0$. Then the exact controllability can be obtained in the same fashion as the finite-dimensional case (Theorem 2.1). For the mathematical justification of this, the readers are referred to ([10]).

3. ENERGY DECAY ESTIMATES FOR THE WAVE EQUATION WITH DISSIPATION

Let us return to the systems (2) or (1) (with (3), (4) or (5)) in H_1. We first define an "energy inner product" on H_1:

$$\left\langle \begin{bmatrix} w_1 \\ v_1 \end{bmatrix}, \begin{bmatrix} w_2 \\ v_2 \end{bmatrix} \right\rangle_E \equiv \int [\text{grad } w_1 \cdot \text{grad } \overline{w}_2 + v_1 \cdot \overline{v}_2]dx.$$

This inner product induces a semi-norm on H_1. By (generalized) Poincaré inequality, this semi-norm is a norm. The energy function for $(w(\cdot, t), v(\cdot, t))$ becomes

$$E(w(\cdot, t), v(\cdot, t)) = \left\langle \begin{bmatrix} w(\cdot, t) \\ v(\cdot, t) \end{bmatrix}, \begin{bmatrix} w(\cdot, t) \\ v(\cdot, t) \end{bmatrix} \right\rangle_E .$$

Theorem 3.1 ([5]) Let w(x, t) be the solution of

$$\begin{cases} \dfrac{\partial^2 w}{\partial t^2} + \alpha(x) \dfrac{\partial w}{\partial t} - \Delta w = 0 \\ w(x, t)|_\Gamma = 0 \\ w(x, 0) = w_0(x) \in H_0^1(\Omega) \\ v(x, 0) \equiv w_t(x, 0) = v_0(x) \in H^0(\Omega) \end{cases} \tag{18}$$

where $\alpha \in L^\infty(\Omega)$ with ess. sup. $\alpha(x) \geq \alpha_0 > 0$. Then

(a) there exists a strongly continuous semigroup S(t) of contractions on H_1 such that $S(t)(w_0, v_0)$ is the solution of the abstract Cauchy problem

$$\frac{d}{dt}\begin{bmatrix} w \\ v \end{bmatrix} = \begin{bmatrix} 0 & I \\ \Delta & -\alpha \end{bmatrix}\begin{bmatrix} w \\ v \end{bmatrix} \tag{19}$$

(b) the energy decays uniformly exponentially, i.e., there exist K, $\delta > 0$ such that

$$\|S(t)\|^2_{L(H_1,\, H_1)} \leq Ke^{-\delta t} \quad \text{for all} \quad t \geq 0. \tag{20}$$

Part (a) is proved by using the Lumer-Phillips theorem ([9]) for dissipative operators. Part (b) is proved by the energy method ([14]).

Combining Theorem 3.1 with Russell's "controllability via stabilizability" principle, we conclude

Theorem 3.2 (Chen and Russell [5]) For given duration $T > 0$ long enough, there exists a control $f \in L^2(\Omega \times [0, T])$ which solves the controllability problem. Furthermore, f can be chosen to lie in $C([0, T], L^2(\Omega))$.

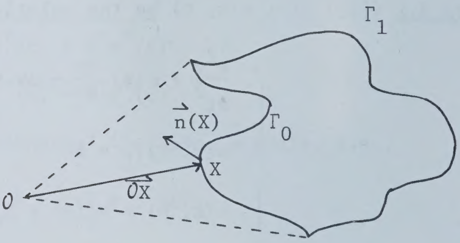
As for the system (1) with (3), (4) or (5), we have the following energy decay and controllability theorems (Theorem 3.4 and 3.5, resp.). They depend strongly on the geometry of $(\Omega, \Gamma_0, \Gamma_1)$.

Definition 3.3: By a "(star-complemented)-(strongly star-shaped)" domain $(\Omega, \Gamma_0, \Gamma_1)$, in short, scsssd, we mean that there exists a point $\mathcal{O}$ (origin) in $\mathbb{R}^N$ such that

$$\overrightarrow{\mathcal{O}X} \cdot \vec{n}(X) \leq 0 \qquad \text{for } X \in \Gamma_0$$

and

$$\overrightarrow{\mathcal{O}X} \cdot \vec{n}(X) \geq \beta > 0 \qquad \text{for } X \in \Gamma_1$$



Theorem 3.4 ([3], [6]) Let $w(x, t)$ be the solution of the wave equation

$$\begin{cases} \dfrac{\partial^2 w}{\partial t^2} - \Delta w = 0 \\ \text{boundary condition} \begin{cases} w(x, t)|_{\Gamma_0} = 0 \\ [\alpha_1 w(x, t) + \alpha_2 \dfrac{\partial w}{\partial n}(x, t) + \alpha_3 \dfrac{\partial w}{\partial t}(x, t)]|_{\Gamma_1} = 0 \end{cases} \\ \text{initial condition} \begin{cases} w(x, 0) = w_0(x) \in H^1_{\Gamma_0}(\Omega) \\ v(x, 0) = w_t(x, 0) = v_0(x) \in H^0(\Omega). \end{cases} \end{cases}$$

where $\alpha_1 \geq 0$, α_2, $\alpha_3 > 0$. Then

(c) there exists a strongly continuous semigroup $S(t)$ of contractions such that $S(t)(w_0, v_0)$ is the solution of

$$\frac{d}{dt}\begin{bmatrix} w \\ v \end{bmatrix} = \begin{bmatrix} 0 & I \\ \Delta & 0 \end{bmatrix}\begin{bmatrix} w \\ v \end{bmatrix}$$

(d) there exist $\overline{T}$, $K > 0$ such that the energy of the solution decays uniformly with a rate K/t^2 for $t \geq \overline{T}$, i.e.,

$$\|S(t)\|^2_{L(H_1, H_1)} \leq \frac{K}{t^2}$$

provided that $(\Omega, \Gamma_0, \Gamma_1)$ forms a scsssd.

Part (c) is also proved by the Lumer-Phillips theorem. The proof for part (d) is considerably more difficult than its counterpart (b). It is done through a modification of the energy invariant method developed by C. S. Morawitz. Theorem 3.4 gives the closest result to Russell's conjecture ([13]).

Theorem 3.5 (Chen [6]) Let $(\Omega, \Gamma_0, \Gamma_1)$ form an scsssd. Then there exists a boundary value control (of Dirichlet, Neumann, or Robin type) $f \in L^2(\Gamma_1 \times [0, T])$ which solves the controllability problem.

Finally we remark that the "controllability via stabilizability" principle can be applied to Schrödinger equations in a bounded domain ([4]). It can also be applied to obtain controllability for the wave equation in a compact Riemannian manifold (Millman and Chen [8]).

REFERENCES

[1] G. Chen, Energy decay estimates and control theory for the wave equation in a bounded domain, doctoral dissertation, University of Wisconsin-Madison, 1977.

[2] G. Chen, Energy decay estimates and exact boundary value controllability for the wave equation in a bounded domain, Part I, in preparation.

[3] __________, Energy decay estimates and exact boundary value controllability for the wave equation in a bounded domain, Part II, in preparation.

[4] G. Chen, Control theory for the Schrödinger equation in a bounded domain, in preparation.

[5] G. Chen and D. L. Russell, Control and stabilization for the wave equation in a bounded domain, in preparation.

[6] G. Chen and D. L. Russell, Asymptotic stability and energy decay rates for the wave equation with boundary dissipation, in preparation.

[7] K. D. Graham and D. L. Russell, Boundary value control of the wave equation in a spherical region, SIAM J. Control, Vol. 13, No. 1, Jan., 1975, 174-196.

[8] R. Millman and G. Chen, Control theory for the wave equation in a compact Riemannian manifold, in preparation.

[9] A. Pazy, Semigroup of linear operators and applications to partial differential equations, Lecture note #10, Dept. of Math., University of Maryland, College Park, 1974.

[10] J. Quinn and D. L. Russell, Asymptotic stability and energy decay rates for solutions of hyperbolic equations with boundary damping, MRC Technical Summary Report #1575, University of Wisconsin-Madison, Nov., 1975.

[11] D. L. Russell, Exact boundary value controllability theorems for wave and heat processes in star-complemented regions, in "Differential Games and Control Theory," Roxin, Liu, Sternberg, Eds., Marcel Dekker, Inc., New York, 1974.

[12] D. L. Russell, Nonharmonic Fourier series in the control of distributed parameter systems, J. Math. Anal. & Appl., Vol. 18 (1967), 542-560.

[13] D. L. Russell, Controllability and stabilizability theory for linear partial differential equations: recent progress and open questions, MRC Technical Summary Report #1700, University of Wisconsin-Madison, Nov., 1976.

[14] W. A. Strauss, The Energy Method in Nonlinear Partial Differential Equations, Instituto de Mathematica Pura e Applicada, Brazil, 1969.

[15] R. Triggiani, Extensions of rank conditions for controllability and observability to Banach spaces and unbounded operators, SIAM J. Control, Vol. 14, No. 2, Feb., 1976, 313-338.

SOME PROBLEMS IN COMMUNICATION NETWORKS

P. Papantoni-Kazakos
Bell Laboratories

ABSTRACT

A general discussion is presented on some of the open problems in communication networks. Routing structures and sources of unsuccessful communication through the network are emphasized. Some open problems involving sophisticated parametric as well as robust statistical algorithms are stated.

Work done at Rice University and supported by the Air Force Grant AFOSR 77-3156.

I. DESCRIPTION OF THE NETWORK

To understand some of the problem involved in reliably communicating messages within the network, some basic network operations must be described.

The smallest element (that is of any interest to the network analyst) in a communication network is a center. A center consists of several units that communicate directly with each other. Different centers communicate through a number of routes, where each time the route one particular message is carried on is chosen hierarchically. Each route consists of a number of links that are, in general, connected to each other through tandems (switching offices). Finally, each link consists of a number of single message carriers that are called trunks, while the tandems connect several centers. A message originating at center A (Figure 1) and with destination another center B follows a routing hierarchy described as follows:

It first tries the direct route that consists of a single link connecting the two centers(dotted line in Figure 1). If all the trunks in this link are functioning properly but are busy, the message tries the next route in the hierarchy (route through A, T_1,B in Figure 1). If this route is also functioning but busy, the message tries the next route in hierarchy and so on, until it reaches the final route available to it (route AT_2T_3 B in Figure 1). If this last route is busy or malfunctioning the message fails to go through and a communication failure to B is recorded at A.

The rejection of the message by a particular route due to full occupancy (at the moment) of all trunks involved is called blocking.

Under healthy network conditions blocking probabilities can be assigned to each route that correspond to a particular center pair (A,B) and are functions of the A to B communication load, the number of routes connecting A,B, and the number of trunks in each such route.
Suppose now that a "average load" time period is considered and the communication from center A to center B is studied. If in some of the routes between A and B a link is malfunctioning (due to some faulty trunk), and if center A is unaware of the malfunction, messages from A to B will keep trying this link with probability specified by the initial routing structure and the "averaging load". As a result to that some messages will be killed by the malfunctioning link and some communication failures from A to B will be recorded. Therefore, in the presence of faulty links which center A is unaware of, communication failures will be caused that are not just due to overload and are not happening just at the highestlevel in hierarchy route.

The routing structure desribed above is based on a trade off between economy and communication efficiency. The direct links (dotted line in Figure 1) carry usually the highest portion of the message load while the higher in hierarchy alternate routes are used during traffic peaks and they have capacity high enough to secure good communication where such peaks are occuring and low enough so that they do not remain idle most of the time.

The performance of the network, as viewed by the users, is measured through its ability to successfully respond to communication attempts. Its efficiency as viewed by an outside observer is a combination of two factors: effectiveness in responding to communication demands, and average degree of occupancy.

II. SOME OPEN PROBLEMS

We are concentrating here on the performance evaluation of the network. The following major question arises in this case:

Is it possible to evaluate the network performance at a particular time, if yes what kind of data are required and how can such an evaluation be effective without utilizing an excessive amount of information? Also, how can malfunctions be localized or even predicted with the use of economically attractive methods?

In two Bell Labs technical memoranda that have not been cleared for publication yet, the author analyzes the use of limited center-to-center successful and unsuccessful communication completions for locating the faulty links that cause the failures, and for continuously monitoring the quality in communication throughout the network. The algorithms used are economical not only because they only utilize a limited amount of information, but also because they use one step memory and are computationally efficient. However, assumptions as to the routing structures have been made that in some cases need relaxation. Specifically, "average load" time periods are observed and the routing probabilities are then considered unchanged. But, even during such time periods the load fluctuations may be momentarily substantial, in which case "robust" algorithms that are mostly insensitive to such load variations must be developed.

Furthermore, the effect of malfunctioning network links to the relationship between communication messages through them must be studied further. Specifically, such links may cause"partial message killing" as well as interference between messages. These effects result in additional reduction of the communication quality within the network.

REFERENCES

[1] P. Papantoni-Kazakos, "Use of Link Call Completions in Quality Control" Bell Laboratories Technical Memorandum in typing process.

[2] P. Papantoni-Kazakos, "The Potential of End-to-End Call Completion Measurements in Network Trouble Localization and Quality Control", Bell Laboratories Technical Memorandum in process.

[3] J. S. Kaufman, "Faulty-Trunk Detection Algorithm Using EADAS/ICUR Traffic Data", Bell System Technical Journal, Vol. 56, no. 6, pp 919-976.

[4] D. Kazakos, "Recursive Estimation of Prior Probabilities Using a Mixture", IEEE Trans. Inform. Theory, Vol. IT-23(2), pp 203-211.

[5] P. Papantoni-Kazakos, "Some Performance Criteria Incorporations Data Dependence in Robust Estimation", Bell Laboratories Technical Memorandum TM-77-3452-4, July 12, 1977.

[6] E. S. Page, "Continuous Inspection Schemes" Biometrika 41, pp 100-115, 1954.

[7] H. Chernoff, "Sequential Analysis and Optimal Design", Regional Conference Series in Applied Mathematics, Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvannia.

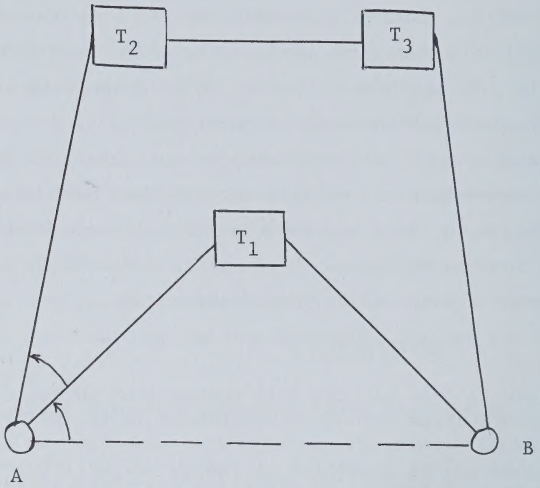


FIGURE I

ROBUST DETECTION OF STOCHASTIC SIGNALS

H.V. POOR
Dept. of Electrical Eng.
and the Coordinated Science Lab.
University of Illinois
Urbana, Illinois 61801

J.B. THOMAS
Dept. of Electrical Eng.
Princeton University
Princeton, New Jersey 08540

ABSTRACT

The design of robust detectors for the discrete-time detection of purely stochastic signals is considered. Two small-signal models for stochastic signals are employed, and a mixture density class is used to represent unknown deviations in noise statistics. By considering known results for deterministic signals, detectors are derived which are insensitive to these types of deviations in the large-sample-size case.

I. INTRODUCTION

In Ref. [1], two models are considered for the detection of purely stochastic signals in noise. One of these models is based on a standard additive-noise assumption, and the other is based on the representation of signal presence by a scale-change in the noise-only statistics. Both of these models provide intuitively reasonable characterizations of the local, or small-signal, case and optimal detectors corresponding to each of these models are derived in [1]. The design of these optimal detectors depends on a complete knowledge of the noise statistics. Unfortunately, there is often some degree of uncertainty concerning the statistical description of noise in detection systems, and thus we are interested in designing systems which are robust or insensitive to perturbations in the noise model.

In this paper, we consider the problem of designing robust detectors for the stochastic-signal problem as formulated in Ref. [1]. This topic has been investigated in part in Ref. [2] for the situation where the noise density is modeled by a mixture of Gaussian densities. In Refs. [3] and [4], the robust detection of known (i.e., deterministic) signals in noise has been considered. In these works, unknown perturbations in noise statistics are modeled by a Huber-Tukey mixture model; that is, by a nominal density mixed with an unknown contaminating density [5]. In this paper we will use a similar model for noise uncertainty, and, restricting our study to the asymptotic, small-signal case, will derive stochastic-signal detectors which exhibit robustness properties similar to those of the procedures established in [3] and [4].

The fundamentals and background for this problem are introduced in Section II by a review of the results of Kassam and Thomas [4] for the robust detection of deterministic signals. In Section III we consider the problem of robust detection under the scale-change model, and it is seen that, for this case, the established theory can be modified to give a solution. In Section IV we consider the additive-noise model and derive a robust detector for this case based on the concepts of the previous two sections.

II. ROBUST DETECTION OF KNOWN SIGNALS

First we consider the problem of detecting known signals in additive white noise. Specifically, we assume a sequence $\{x_i\}_{i=1}^{n} \equiv \underline{x}$ of real observations of a process $\{X_i\}_{i=1}^{n} \equiv \underline{X}$, and, on the basis of $\underline{x}$, we wish to decide whether or not a known signal is present in $\underline{X}$. We have the hypothesis pair

$$\begin{aligned} &H_0: X_i = N_i \qquad ; \; i = 1,\dots,n \\ \text{versus} \\ &H_1: X_i = N_i + \theta s_i \; ; \; i = 1,\dots,n \end{aligned} \tag{1}$$

where $\{N_i\}_{i=1}^{n}$ is an independent, identically-distributed noise sequence with common univariate density f; the $\{s_i\}_{i=1}^{n}$ is a known signal sequence; θ is a positive signal-strength parameter. We assume that $0 < e_1 \le |s_i| \le e_2 < \infty$ for all i and some e_1 and e_2; and, for the local case, consider the limit as $\theta \to 0^+$.

If the noise density f is known and sufficiently regular, then a size-α locally-optimum detector for the problem of Eq. (1) is given by [6]

$$\phi_{\ell o}(\underline{x}) = \begin{cases} 1 & \\ \gamma & \text{if } \sum_{i=1}^{n} s_i g_{\ell o}(x_i) \\ 0 & \end{cases} \begin{cases} > \tau \\ = \tau \\ < \tau \end{cases} \tag{2}$$

where $g_{\ell o} \overset{\Delta}{=} -f'/f$, the function $\phi_{\ell o}(\underline{x})$ is the probability with which we accept H_1 given that $\underline{x}$ is observed, and γ and τ are chosen to give size α. By locally-optimum we mean that this test maximizes the slope of the power curve $\partial E_\theta\{\phi\}/\partial\theta$, at $\theta=0$, among all possible tests ϕ satisfying $E_0\{\phi\} \le \alpha$, the pre-assigned level[7]

In [4], Kassam and Thomas have studied the problem of designing a robust detector for the hypothesis pair of Eq. (1). The following is a summary of their results:

We assume that the noise density f is known only to be a member of a Huber-Tukey mixture class $\mathfrak{F}(\epsilon, f_0, H)$ given by

$$\mathfrak{F}(\epsilon, f_0, H) = \{f: f = (1-\epsilon)f_0 + \epsilon h; \; h \epsilon H\} \tag{3}$$

where H is a wide class of density functions; f_0 is a known density; and ϵ is a fixed number between zero and one. The density f_0 can be thought of as a nominal model, h as an unknown "contaminating" density, and ϵ as a measure of the degree of uncertainty or contamination in the model.

A robust size-α detector ϕ_R for H_0 versus H_1 for the class $\mathfrak{F}(\epsilon, f_0, H)$ is defined in [3] or [4] as one which satisfies asymptotically ($n \to \infty$) the conditions:

$$E_0\{\phi_R | f\} \le E_0\{\phi_R | f_R\} = \alpha \tag{4}$$

and

$$\frac{\partial E_\theta\{\phi_R|f\}/\partial\theta|_{\theta=0}}{\partial E_\theta\{\phi_R|f_R\}/\partial\theta|_{\theta=0}} \geq 1 \geq \frac{\partial E_\theta\{\phi|f_R\}/\partial\theta|_{\theta=0}}{\partial E_\theta\{\phi_R|f_R\}/\partial\theta|_{\theta=0}} \tag{5}$$

for all $f\in\mathfrak{F}(\epsilon,f_0,H)$, all α-level tests of H_0 versus H_1, and for some "worst-case" density $f_R\in\mathfrak{F}(\epsilon,f_0,H)$. Here $E_\theta\{\phi|f\}$ denotes the expectation of ϕ when the signal strength is θ and the true noise density is f. Note that the right side of Eq.(5) requires that ϕ_R be the locally-optimum α-level test [as in Eq.(2)] for Eq.(1) when $f\equiv f_R$, so that the problem of finding ϕ_R is equivalent to that of finding f_R.

To find a solution to Eqs.(4) and (5), we place the following conditions on the density f_0:

A1: f_0 is symmetric with support $(-\infty,\infty)$.

A2: f_0 is strongly unimodal; i.e., $-\log f_0$ is strictly convex on $(-\infty,\infty)$.

A3: f_0 has a derivative f_0' which is absolutely continuous on $(-\infty,\infty)$.

We also assume:

A4: The class H is the set of all bounded, symmetric density functions.

Based on these assumptions, we have the following result from [4]:

Theorem 1 (Kassam and Thomas): Suppose that f_0 satisfies A1 through A3 and that H is as in A4. Then, for the problem of Eq. (1) with $f\in\mathfrak{F}(\epsilon,f_0,H)$, a robust detector exists in the sense of Eqs.(4) and (5). It is the locally-optimum test [as in Eq.(2)] corresponding to the noise density f_R given by

$$f_R(x) = \begin{cases} (1-\epsilon)f_0(x) & ; |x| \leq a \\ (1-\epsilon)f_0(a)e^{-b||x|-a|} & ; |x| > a \end{cases} \tag{6}$$

where a and b are chosen so that $-f_0'(a)/f_0(a) = b$ and $\int f_R = 1$. Eq.(5) holds provided that the size $\alpha \geq \alpha_\epsilon$, a lower bound depending only on ϵ and f_0; and Eq.(4) holds for any α between 0 and 1.

This theorem states that the robust test ϕ_R is given by

$$\phi_R(\underline{x}) = \begin{cases} 1 \\ \gamma' \\ 0 \end{cases} \text{if } \sum_{i=1}^{n} s_i g_R(x_i) \begin{cases} > \tau' \\ = \tau' \\ < \tau' \end{cases} \tag{7}$$

where

$$g_R(x) = -f_R'(x)/f_R(x) = \begin{cases} -f_0'(x)/f_0(x) & ; |x| \leq a \\ b\,\mathrm{sgn}(x) & ; |x| > a \end{cases} \tag{8}$$

Thus, the test $\phi_{\ell o}$ based on the assumptions $f \equiv f_0$ is "robustified" by the introduction of limiting for observations whose magnitudes are greater than a. Further details of this result may be found in [4] or [8].

III. ROBUST DETECTION UNDER THE SCALE-CHANGE MODEL

We now would like to have a result like Theorem 1 for the case of detecting stochastic signals as modeled in Ref. [1]. To study this, we first consider the scale-change model of Ref. [1]; that is, we assume as before that we have a sequence $\{x_i\}_{i=1}^{n} \equiv \underline{x}$ of independent observations of a process $\{X_i\}_{i=1}^{n} \equiv \underline{X}$ and we wish to decide between the following pair of hypotheses:

$$\begin{array}{ll} & H_0\text{: } X_i \text{ has density } f(x) \qquad ;\ i = 1,\ldots,n \\ \text{versus} & \\ & H_\sigma\text{: } X_i \text{ has density } f(x/\sigma)/\sigma\ ;\ i = 1,\ldots,n \end{array} \qquad (9)$$

where f is the noise-only density, and the scale parameter σ is greater than unity (for simplicity we consider the time-invariant case). The local, or small-signal, case corresponds to the limit as $\sigma \to 1^+$.

If the noise density f is known and sufficiently regular, then a size-α locally-optimum detector for Eq. (9) is given by [1],

$$\phi_\sigma(\underline{x}) = \left\{\begin{matrix} 1 \\ \nu \\ 0 \end{matrix}\right. \text{ if } \sum_{i=1}^{n} g_\sigma(x_i) \left\{\begin{matrix} > \mu \\ = \mu \\ < \mu \end{matrix}\right. \qquad (10)$$

where $g_\sigma(x) = -\,(xf'(x)/f(x) - 1)$. Here, ν and μ are chosen to give size α; and ϕ_σ maximizes the slope of the power curve, $\partial E_\sigma\{\phi\}/\partial\sigma$, at $\sigma = 1$, among all α-level tests ϕ of H_0 versus H_σ.

As in the known-signal case of Section II, we wish to consider the situation where f is known only to be a member of a Huber-Tukey class $\mathfrak{F}(\epsilon, f_0, H)$ as in Eq. (3). We will restrict H to consist only of bounded symmetric densities, and will place further restrictions on H in the following paragraphs.

Analogously to Eqs. (4) and (5), we define a size-α robust test $\phi_{\sigma,R}$ for H_0 versus H_σ for the class $\mathfrak{F}(\epsilon, f_0, H)$ as one which satisfies asymptotically ($n \to \infty$) the conditions

$$E_0\{\phi_{\sigma,R} | f\} \leq E_0\{\phi_{\sigma,R} | f_{\sigma,R}\} = \alpha \qquad (11)$$

and

$$\frac{\partial E_\sigma\{\phi_{\sigma,R} | f\}/\partial\sigma\big|_{\sigma=1}}{\partial E_\sigma\{\phi_{\sigma,R} | f_{\sigma,R}\}/\partial\sigma\big|_{\sigma=1}} \geq 1 \geq \frac{\partial E_\sigma\{\phi | f_{\sigma,R}\}/\partial\sigma\big|_{\sigma=1}}{\partial E_\sigma\{\phi_{\sigma,R} | f_{\sigma,R}\}/\partial\sigma\big|_{\sigma=1}} \qquad (12)$$

for all $f \in \mathfrak{F}(\epsilon, f_0, H)$, all α-level tests ϕ of H_0 versus H_σ and for some particular "worst-case" density $f_{\sigma,R} \in \mathfrak{F}(\epsilon, f_0, H)$. Here $E_\sigma\{\phi | f\}$ denotes the expectation under H_σ of the test ϕ when σ

is the scale parameter and f is the true noise density.

To look for a solution to Eqs.(11) and (12) we first consider the following heuristic argument:

Due to the symmetry of the noise density f, the algebraic signs of the observations x_i are irrelevant in the detection of a scale change. Thus, we could observe the sequence $\{y_i\}_{i=1}^{n} = \{\log|x_i|\}_{i=1}^{n}$ rather than $\{x_i\}_{i=1}^{n}$ without losing any information for the detection problem of Eq.(9); and the hypothesis pair of Eq.(9) is equivalent to the pair

$$\begin{array}{ll} & H_0: Y_i \text{ has density } k(y) \quad ; \ i = 1,\dots,n \\ \text{versus} & \\ & H_\sigma: Y_i \text{ has density } k(y-\log\sigma) \ ; \ i = 1,\dots,n \end{array} \qquad (13)$$

where $Y_i = \log|X_i|$ and $k(y) = e^y f(e^y)u(y)$ is the density of $Y = \log|X|$ when X has density f. [Here u(y) is the unit step: $u(y) = 1$ if $y \geq 0$; $u(y) = 0$ if $y < 0$].

Comparing Eq.(13) to Eq.(1) we see that Eq.(13) is of the form of the known-signal problem with noise density k(y) and constant signal $\log(\sigma)$. Note that if $f \in \mathfrak{J}(\epsilon, f_0, H)$ then $k \in \mathfrak{J}(\epsilon, k_0, \mathcal{L})$ where $k_0(y) = e^y f_0(e^y)u(y)$, and $\mathcal{L}$ is the class

$$\mathcal{L} = \{\ell : \ell(y) = e^y h(e^y)u(y) \ ; \ h \in H\} \qquad (14)$$

We also note that, with $\theta \triangleq (\sigma - 1)$, we have

$$\log\sigma = \theta + o(\theta)$$

so that, in the analysis of the detection problem of Eq.(13), derivatives with respect to σ as σ approaches unity may be replaced with derivatives with respect to θ as θ approaches zero; under this transformation, Eq.(13) is equivalent to

$$\begin{array}{ll} & H_0: Y_i \text{ has density } k(y) \quad ; \ i = 1,\dots,n \\ \text{versus} & \\ & H_1: Y_i \text{ has density } k(y-\theta) \ ; \ i = 1,\dots,n \end{array} \qquad (15)$$

as θ approaches zero.

Thus, the problem of designing a detector to be robust over $\mathfrak{J}(\epsilon, f_0, H)$ in the stochastic-signal sense of Eqs.(11) and (12) can be transformed into the problem of designing a robust detector over the class $\mathfrak{J}(\epsilon, k_0, \mathcal{L})$ in the known-signal sense of Eqs.(4) and (5). With this in mind we state the following result:

Consider the following conditions:

B1: The density f_0 is symmetric with support $(-\infty, \infty)$.

B2: The density $k_0(y) = e^y f_0(e^y)u(y)$ is strongly unimodal on $(0, \infty)$. [This is true, for example, if $f_0(x)$ is strongly unimodal on $(-\infty, \infty)$].

B3: The derivative of k_0 is absolutely continuous on $(0, \infty)$.

B4: There are numbers $c > 0$ and $d > 1$ satisfying

$$(1-\epsilon)^{-1} = \int_{-c}^{c} f_0(x)dx + 2cf_0(c)/(d-1)$$

and $\qquad$ (16)

$$d = -\, cf_0'(c)/f_0(c)$$

(For given f_0, this places an upper bound on the parameter ϵ).

B5: The class H is the sub-class of all bounded symmetric densities h which satisfy

$$\int_{-\infty}^{\infty} [(1-\epsilon)f_0(x) + \epsilon h(x)]g_{\sigma,R}(x)dx = 0 \tag{17}$$

where

$$g_{\sigma,R}(x) = \begin{cases} -xf_0'(x)/f_0(x) - 1 \; ; & |x| \leq c \\ d - 1 \; ; & |x| > c \end{cases} \tag{18}$$

and c and d are from Eq. (16). We then have a theorem analogous to Theorem 1:

Theorem 2: Suppose that f_0 and ϵ satisfy conditions B1 through B4, and that H satisfies B5. Then for the scale-change problem of Eq. (9) with $f \epsilon \mathfrak{F}(\epsilon, f_0, H)$, a robust detector exits in the sense of Eq. (11) and (12). It is the locally-optimum detector $\phi_{\sigma,R}$ [as in Eq. (10)] corresponding to the noise density $f_{\sigma,R}$ given by

$$f_{\sigma,R}(x) = \begin{cases} (1-\epsilon)f_0(x) \; ; & |x| \leq c \\ (1-\epsilon)f_0(c)(|x|/c)^{-d} \; ; & |x| > c \end{cases} \tag{19}$$

where c and d are the solutions to Eq. (16). Furthermore, for this detector we have

$$E_0\{\phi_{\sigma,R} | f\} \equiv E_0\{\phi_{\sigma,R} | f_{\sigma,R}\} = \alpha \tag{20}$$

and

$$\frac{\partial E_\sigma\{\phi_{\sigma,R}| f\}/\partial\sigma\Big|_{\sigma=1}}{\partial E_\sigma\{\phi_{\sigma,R}| f_{\sigma,R}\}/\partial\sigma\Big|_{\sigma=1}} \equiv 1 \geq \frac{\partial E_\sigma\{\phi| f_{\sigma,R}\}/\partial\sigma\Big|_{\sigma=1}}{\partial E_\sigma\{\phi_{\sigma,R}| f_{\sigma,R}\}/\partial\sigma\Big|_{\sigma=1}} \tag{21}$$

for all $f \epsilon \mathfrak{F}(\epsilon, f_0, H)$. Equations (20) and (21) hold for any choice of α between zero and one.

The proof of this theorem is similar to that of Theorem 1, and will be omitted. Note that $g_{\sigma,R}$ of Eq. (18) is the optimum nonlinearity [as in Eq. (10)] corresponding to the noise density $f_{\sigma,R}$ of Eq. (19); that is,

$$g_{\sigma,R}(x) = -\, xf_{\sigma,R}'(x)/f_{\sigma,R}(x) - 1$$

Thus, we have that, for this case, the robust detector $\phi_{\sigma,R}$ is given by

$$\phi_{\sigma,R}(\underline{x}) = \begin{cases} 1 \\ \nu' \\ 0 \end{cases} \text{if } \sum_{i=1}^{n} g_{\sigma,R}(x_i) \begin{cases} > \mu' \\ = \mu' \\ < \mu' \end{cases} \tag{22}$$

As in the known signal case, the locally-optimum detector ϕ_σ based on the assumption $f = f_0$ is robustified by the introduction of limiting in the tails.

Heuristic justification for the 'worst-case' density $f_{\sigma,R}$ of Eq.(19) is given if we note that the density in the transformed class $\mathfrak{F}(\epsilon,k_0,\mathcal{L})$ corresponding to $f_{\sigma,R}$ is given by

$$k_{\sigma,R}(y)=e^y f_{\sigma,R}(e^y)u(y)=\begin{cases}(1-\epsilon)k_0(y) & ;\ 0 \leq y \leq \log c\\ (1-\epsilon)k_0(\log c)\exp\{-(d-1)(y-\log c)\} & ;\ y > \log c\end{cases} \tag{23}$$

Comparing Eq.(23) to the known-signal worst-case density f_R of Eq.(6), we see that, by transforming the scale-change problem to the known-signal problem in Eq.(15), we may infer the robust solution for the scale-change model from Theorem 1. Unfortunately, Theorem 1 can not be applied directly since the densities in $\mathfrak{F}(\epsilon,k_0,\mathcal{L})$ are one-sided and thus do not satisfy conditions A1 and A4. Conditions B1, B4 and B5 are designed to compensate for this asymmetry.

The over-riding result of Theorem 2 is the fact that the asymptotic performance of $\phi_{\sigma,R}$ is constant over the entire class $\mathfrak{F}(\epsilon,f_0,H)$. This result may be surprising at first glance; however, it is due to the fact that the class H satisfying B5 is more special (i.e., narrower) than it appears from its definition. We have the following result concerning this class:
Lemma 1: If f_0 and ϵ satisfy Conditions B1 through B4, then the class H defined by Condition B5 is equivalent to the class of all bounded symmetric densities whose supports lie exclusively outside the interval $(-c,c)$.

The proof of this lemma is straightforward. It depends primarily on the strong unimodality of the density k_0 of Condition B2. We see from this result that the detector $\phi_{\sigma,R}$ is robust over a class of noise densities which are known on $(-c,c)$ and are relatively arbitrary elsewhere. This is a restrictive class; however, this is frequently the type of contamination with which we are concerned, and the resulting detector $\phi_{\sigma,R}$ is an intuitively reasonable one.* Note that it follows from Lemma 1 that the distribution of the statistic $\sum_{i=1}^{n} g_{\sigma,R}(x_i)$ of Eq.(22) is invariant over the class $\mathfrak{F}(\epsilon,f_0,H)$ under the noise-only hypothesis H_0. Thus, the result $E_0\{\phi_{\sigma,R}|f\} \equiv E_0\{\phi_{\sigma,R}|f_{\sigma,R}\}$ of Eq.(20) is true even for finite sample size.

---

*A similar result has been studied by Huber [5] for the problem of robust estimation of a scale parameter in contaminated Gaussian noise.

IV. ROBUST DETECTION UNDER THE ADDITIVE-NOISE MODEL

We now consider the additive-noise model of Ref. [1] for detecting stochastic signals; that is, we assume that we again have a sequence $\{x_i\}_{i=1}^n$ of independent observations of a process $\{X_i\}_{i=1}^n$ and we wish to decide between the hypotheses

$$\begin{aligned} &H_0: X_i = N_i \qquad ; i = 1,\ldots,n \\ \text{versus} \qquad & \\ &H_\theta: X_i = N_i + \theta^{1/2} S_i \ ; i = 1,\ldots,n \end{aligned} \tag{24}$$

where $\{N_i\}_{i=1}^n$ is an independent, identically-distributed noise sequence with common univariate density f; the sequence $\{S_i\}_{i=1}^n$ is an independent, zero-mean signal sequence with known variances $\{\sigma_i^2\}_{i=1}^n$ and (possibly unknown) distributions $\{G_i\}_{i=1}^n$; and θ is a positive number. We assume that $\{N_i\}_{i=1}^n$ is independent of $\{S_i\}_{i=1}^n$, and, for the local case, take the limit as $\theta \to 0^+$. We also assume that $0 < e_3 \leq \sigma_i^2 \leq e_4 < \infty$ for all i and for some e_3 and e_4.

If the noise density f is known, and if f and the signal distributions $\{G_i\}_{i=1}^n$ are sufficiently regular, then a size-α locally-optimum detector for Eq. (24) is given by [1]

$$\phi_\theta(\underline{x}) = \begin{cases} 1 & \\ \delta & \text{if } \left(\sum_{i=1}^n \sigma_i^2 g_\theta(x_i) \right) / 2 \\ 0 & \end{cases} \begin{matrix} > \xi \\ = \xi \\ < \xi \end{matrix} \tag{25}$$

where $g_\theta(x) = f''(x)/f(x)$. Here, as before, δ and ξ are chosen to give size-α; and ϕ_θ maximizes the slope of the power curve, $\partial E_\theta\{\phi\}/\partial\theta$, at $\theta = 0$, among all α-level tests of H_0 versus H_θ.

As in the previous two sections, we consider the situation where the noise density f is known only to be a member of a Huber-Tukey class $\mathfrak{F}(\epsilon, f_0, H)$.* An α-level robust detector $\phi_{\theta,R}$ for this case can be defined analogously to Eqs. (4) and (5) or Eqs. (11) and (12) as one which satisfies asymptotically

$$E_0\{\phi_{\theta,R} | f\} \leq E_0\{\phi_{\theta,R} | f_{\theta,R}\} = \alpha \tag{26}$$

and

$$\frac{\partial E_\theta\{\phi_{\theta,R} | f\}/\partial\theta \big|_{\theta=0}}{\partial E_\theta\{\phi_{\theta,R} | f_{\theta,R}\}/\partial\theta \big|_{\theta=0}} \geq 1 \geq \frac{\partial E_\theta\{\phi | f_{\theta,R}\}/\partial\theta \big|_{\theta=0}}{\partial E_\theta\{\phi_{\theta,R} | f_{\theta,R}\}/\partial\theta \big|_{\theta=0}} \tag{27}$$

for all $f \in \mathfrak{F}(\epsilon, f_0, H)$, all α-level tests ϕ of H_0 versus H_θ, and some $f_{\theta,R} \in \mathfrak{F}(\epsilon, f_0, H)$. Here $E_\theta\{\phi | f\}$ denotes the expectation of

---

*A similar problem has been studied in [2] for the case where both the nominal density f_0 and the contaminating density h are zero-mean Gaussian.

ϕ under the θ-alternative H_θ of Eq. (24) when the true noise density is f.

In both the known-signal and scale-change detection problems, robustness over a class $\mathfrak{F}(\epsilon, f_0, H)$ was achieved by the introduction of limiting in the tails of the locally-optimum nonlinearity corresponding to the assumption $f \equiv f_0$. With this in mind, we propose the following result:

Consider the following conditions concerning the nominal density f_0:

C1: f_0 is symmetric with support $(-\infty, \infty)$.

C2: f_0 is strongly unimodal on $(-\infty, \infty)$.

C3: f_0 has a second derivative f_0'' which is absolutely continuous on $(-\infty, \infty)$.

We also assume

C4: H is the class of all bounded, symmetric densities whose supports lie exclusively outside of the interval $(-\ell, \ell)$, where ℓ is the solution to the equations

$$(1-\epsilon)^{-1} = \int_{-\ell}^{\ell} f_0(x) + 2f_0(\ell)/m$$

and

$$m^2 = f_0''(\ell)/f_0(\ell) \tag{28}$$

Using these conditions we have

Theorem 3: Suppose f_0 satisfies Conditions C1 through C3, and that H satisfies Condition C4. Then for the problem of Eq. (24) with $f \in \mathfrak{F}(\epsilon, f_0, H)$, a robust detector exists in the sense of Eqs. (26) and (27). It is the size-α locally-optimum detector $\phi_{\theta,R}$ [as in Eq. (25)] corresponding to the noise density $f_{\theta,R}$ given by

$$f_{\theta,R}(x) = \begin{cases} (1-\epsilon) f_0(x) & ; \ |x| \leq \ell \\ (1-\epsilon) f_0(\ell) e^{-m||x|-\ell|} & ; \ |x| > \ell \end{cases} \tag{29}$$

where ℓ and m are the solutions to Eqs. (28). Specifically, for this detector we have

$$E_0\{\phi_{\theta,R} | f\} \equiv E_0\{\phi_{\theta,R} | f_{\theta,R}\} = \alpha \tag{30}$$

and

$$\frac{\partial E_\theta\{\phi_{\theta,R} | f\}/\partial\theta \big|_{\theta=0}}{\partial E_\theta\{\phi_{\theta,R} | f_{\theta,R}\}/\partial\theta \big|_{\theta=0}} \equiv 1 \geq \frac{\partial E_\theta\{\phi | f_{\theta,R}\}/\partial\theta \big|_{\theta=0}}{\partial E_\theta\{\phi_{\theta,R} | f_{\theta,R}\}/\partial\theta \big|_{\theta=0}} \tag{31}$$

for all f in $\mathfrak{F}(\epsilon, f_0, H)$ and all α-level tests ϕ of H_0 versus H_θ. Equations (30) and (31) hold for any choice of α between zero and one.

The proof of this theorem is similar to the proof of Theorem 1. Note that the detector $\phi_{\theta,R}$ is given by

$$\phi_{\theta,R}(\underline{x}) = \begin{cases} 1 \\ \delta' \quad \text{if} \left(\sum_{i=1}^{n} \sigma_i^2 g_{\theta,R}(x_i) \right) / 2 \\ 0 \end{cases} \begin{cases} > \xi' \\ = \xi' \\ < \xi' \end{cases} \tag{32}$$

where

$$g_{\theta,R}(x) \triangleq f''_{\theta,R}(x)/f_{\theta,R}(x) = \begin{cases} f''_0(x)/f_0(x) & ; \ |x| \leq \ell \\ m^2 & ; \ |x| > \ell \end{cases} \tag{33}$$

Thus, as in the previous sections, limiting in the tails is the robustifying mechanism for this detector; and we have an intuitively satisfying result despite the restrictions placed on the class H by Condition C4.

V. CONCLUSIONS

We have considered the design of robust detectors for the two stochastic-signal models of Ref. [1]. The robust detectors (as defined here) are characterized by "worst-case" densities for which they are optimum. For the scale-change model, this density is a version of the nominal density with tails modified to decay geometrically. For the additive noise model this density is a version of the nominal density modified to have exponentially-decaying tails. For the stochastic signal problem, no lower bound is needed on the false-alarm probability α as is necessary for the known-signal case.

The solutions given here are not as complete as for the known-signal case of Section II because of the additional restrictions on the density classes. However, the requirement that the uncertainty in the noise statistics be confined to the tails of the density is not overly restrictive since this kind of contamination is probably the most damaging in the problems studied here.

ACKNOWLEDGEMENTS

The authors are grateful for the support of the National Science Foundation under Grant ENG-76-19808 and the U.S. Army Research Office under Grant DAAG29-75-G-0192.

REFERENCES

1. H.V. Poor and J.B. Thomas, "Local detection of stochastic signals in additive noise", Proc. Fourteenth Annual Allerton Conference Circuit and System Theory, pp.432-440, Sept.-Oct. 1976.
2. R.D. Martin and C.P. McGath, "Robust detection of stochastic signals," IEEE Trans. Inform. Theory, Vol. IT-20, pp.537-541, July 1974.
3. R.D. Martin and S.C. Schwartz, "Robust detection of a known signal in nearly-Gaussian noise," IEEE Trans. Inform. Theory, Vol. IT-17, pp.50-56, Jan. 1971.
4. S.A. Kassam and J.B. Thomas, "Asymptotically robust detection of a known signal in contaminated non-Gaussian noise," IEEE Trans. Inform. Theory, Vol. IT-22, pp.22-26, Jan. 1976.
5. P.J. Huber, "Robust estimation of a location parameter," Ann. Math. Stat., Vol. 35, pp.73-101, March 1964.
6. J.H. Miller and J.B. Thomas, "Detectors for discrete-time signals in non-Gaussian noise," IEEE Trans. Inform. Theory, Vol. IT-18, pp.241-250, March 1972.
7. T.S. Ferguson, Mathematical Statistics. New York: Academic,1967.
8. S.A. Kassam, "Nonparametric and robust detection: dependent data and conditional tests," Ph.D. dissertation, Princeton University, Department of Electrical Engineering, Aug. 1975.

TWO-INPUT CORRELATOR DETECTORS FOR RANDOM SIGNALS

JUNG GIL SHIN AND SALEEM A. KASSAM
Department of Systems Engineering
Moore School of Electrical Engineering
University of Pennsylvania
Philadelphia, PA. 19104

ABSTRACT

A multi-level coincidence-correlator detector is considered for two-input random-signal detection. Based on the maximum-efficacy criterion the structure of the optimum detector is derived. Some properties of the optimum detector are examined. Comparisons are made between the performance of the unconstrained optimum detector, the (constrained) optimum quantizer-correlator and the generalized dead-zone limiter coincidence detector.

I. INTRODUCTION

The problem of detecting weak random signals against a background of additive noise arises in many practical situations such as in underwater sound applications, geophysical and astronomical problems. Considering two-input systems, there are several detectors, both parametric and nonparametric, which may be used for these detection problems. In general, parametric detectors do not perform well when the assumption about the underlying noise density is not valid. On the other hand, nonparametric detectors which perform well often require complex implementation.

In this paper we consider a class of correlators which we call multi-level coincidence-correlators (CC). It includes the nonparametric polarity coincidence correlator (PCC) [1] and its extensions such as the dead-zone limiter coincidence (DZC) detector [2]. Basically a multi-level CC is a quantizer which quantizes input data in two-dimensional space. The symmetric 2m-level CC calculates the test statistic T given by

$$T = \sum_{i=1}^{n} \left\{ \sum_{j=1}^{m} Q_j(X_1^i, X_2^i) \operatorname{sgn}(X_1^i) \operatorname{sgn}(X_2^i) \right\} \tag{1}$$

where

$$Q_j(X_1^i, X_2^i) = \begin{cases} w_j \,, & c_j \le g(|X_1^i|, |X_2^i|) < c_{j-1} \\ -w_j \,, & -c_{j-1} < g(|X_1^i|, |X_2^i|) \le -c_j \\ 0 \,, & \text{otherwise} \end{cases} \tag{2}$$

$\underline{c} = (c_0, c_1, \ldots, c_m)$ is a threshold parameter vector and g is a symmetric threshold function. The parameter n is the size of each input vector of observations $\underline{X}_1 = (X_1^1, X_1^2, \ldots, X_1^n)$ and $\underline{X}_2 = (X_2^1, X_2^2, \ldots, X_2^n)$. We can formulate a nonparametric detection scheme based on T by using the idea of conditional

---

This research is supported by the National Science Foundation under grant ENG 76-10444.

testing [3,4]. Also the threshold function g can be selected to result in simplicity of implementation.

Optimality is considered based on the criterion of detection efficacy, an asymptotic performance measure for signal-to-noise ratio (SNR) approaching zero. The optimum 2m-level CC is shown to be the minimum mean-squared-error (MSE) fit to $\ell(x_1^i)\ell(x_2^i)$ where $\ell(x) = -f'(x)/f(x)$, the locally optimum nonlinearity for the noise density f. As an example of a CC detector, a quantizer-correlator (QC) is also considered. In the QC each input is separately quantized prior to correlation.

Numerical performances (in terms of efficacies) for various types of CC detectors will be compared to that of the optimum CC, with the number of levels as a variable. The results show the trade-off between performance and complexity for the various CC detectors.

II. 2M-LEVEL COINCIDENCE-CORRELATORS

Given the two observation vectors $\underline{X}_1$ and $\underline{X}_2$, we want to decide between a null hypothesis H_0 and an alternative H_1 which may be defined by

$$H_0: \quad X_1^i = N_1^i, \qquad X_2^i = N_2^i, \qquad i=1,2,\dots,n, \tag{3}$$

$$H_1: \quad X_1^i = N_1^i + \theta S_i, \quad X_2^i = N_2^i + \theta S_i, \quad i=1,2,\dots,n. \tag{4}$$

The S_i, i=1,2,...,n, representing the random signal are assumed to be of zero-mean and unit-variance. The components of $\underline{N}_1$ and $\underline{N}_2$ (noise) are assumed independent and identically distributed. Let the noise density function be f, which is assumed symmetric about the origin, with corresponding cumulative distribution function denoted by F. If σ^2 is the noise variance, then θ^2/σ^2 is the input SNR.

As we mentioned earlier, the CC quantizes input data in two-dimensional space. Let us define regions $P_1, P_2, \dots, P_m$ and $N_1, N_2, \dots, N_m$ in two-dimensional data space such that

$$\left.\begin{aligned} P_j &= \{(X_1^i, X_2^i):\ c_j \le g(|X_1^i|, |X_2^i|)\,\mathrm{sgn}(X_1^i)\,\mathrm{sgn}(X_2^i) < c_{j-1}\} \\ N_j &= \{(X_1^i, X_2^i):\ -c_{j-1} < g(|X_1^i|, |X_2^i|)\,\mathrm{sgn}(X_1^i)\,\mathrm{sgn}(X_2^i) \le -c_j\}, \end{aligned}\right\} \tag{5}$$

g being some symmetric function for non-negative arguments. The c_j are constants with $c_0 \overset{\Delta}{=} \infty$, $c_m \overset{\Delta}{=} 0$ and $c_j < c_{j-1}$, j=1,2,...,m. Let $\underline{w} = (w_1, w_2, \dots, w_n)$ be the vector of positive weights given to the $\{P_j\}$, w_j to P_j and $-w_j$ to N_j. Figure 1 shows an example of such two-dimensional quantization. Note that we are assuming that the partitioning in each quadrant is identical and the regions are symmetric about the coordinate axes and the main diagonals. This is not an unreasonable assumption since the noise density f is assumed symmetric about the origin and the samples from each channel are independent and should be equally weighted.

Let us denote by $n_{P1}, \dots, n_{Pm}, n_{N1}, \dots, n_{Nm}$ the numbers of sample pairs

(x_1^i, x_2^i) which fall in the regions $P_1,\ldots,P_m$, $N_1,\ldots,N_m$ respectively. The test statistic T can be written as

$$T = \sum_{j=1}^{m} w_j (n_{Pj} - n_{Nj}) \tag{6}$$

Note that this is equivalent to (1). As shown in [3,4] it is possible to formulate a nonparametric detection scheme since the distribution of T conditioned on the n_{cj}, $j=1,\ldots,m-1$, is known, where $n_{cj} \triangleq n_{Pj} + n_{Nj}$. The performance comparison of two detectors will be based on their ARE (asymptotic relative efficiency) [5,6]. Under certain regularity conditions (which will be the case for our detectors), the ARE can be expressed as the ratio of the detector efficacies [7], the efficacy E for the test statistic T being defined as

$$E = \lim_{n\to\infty} \frac{\left\{ \frac{d}{d\theta^2} E_\theta\{T|H_1\}|_{\theta=0} \right\}^2}{n \operatorname{Var}\{T|H_0\}} \tag{7}$$

It has been shown in [4] that for conditional tests similar to those considered here the ARE is generally consistent with finite-sample performance. Thus we consider the efficacy as the detector criterion to be optimized.

From (7), the efficacy E for the test statistic T is expressed as

$$E = \frac{2\left\{ \sum_{j=1}^{m} w_j \iint_{Pj} f'(x) f'(y)\,dxdy \right\}^2}{\sum_{j=1}^{m} w_j^2 \iint_{P_j} f(x) f(y)\,dxdy} \tag{8}$$

The derivation of (8) is simple and is outlined in the Appendix.

Let us now find the vector $\underline{w}^*$ which maximizes the efficacy E. We can obtain $\underline{w}^*$ by finding the local maxima of E wrt w_j, $j=1,2,\ldots,m$. By ignoring a scaling constant (which is immaterial for detection efficacy) we get

$$w_j^* = \frac{\iint_{P_j} f'f'}{\iint_{P_j} ff}, \quad j=1,2,\ldots,m. \tag{9}$$

Here $\iint_{P_j} f'f'$ is the simplified notation for $\iint_{P_j} f'(x)f'(y)dxdy$. Substituting w_j^* into (8) yields

$$E_{\underline{w}^*} = 2 \sum_{j=1}^{m} \frac{\left\{\iint_{P_j} f'f'\right\}^2}{\iint_{P_j} ff} \tag{10}$$

We see from (5) that the regions P_j and N_j, $j=1,2,\ldots,m$, are defined by the threshold parameter vector $\underline{c}$ and the threshold function g. One may restrict the threshold function g to a particular one for some reason, for example, ease of implementation. In such cases where g is constrained, we can find $\underline{c}_g^*$, the threshold parameter of a constrained-optimum CC, by maximizing $E_{\underline{w}^*}$ wrt $\underline{c}$. We can also find the set of regions which maximize $E_{\underline{w}^*}$ when there is no constraint imposed on g. We have the following result.

Theorem 1: The set of regions maximizing $E_{\underline{w}^*}$ of a symmetric, unconstrained 2m-level CC is defined by

$$\left.\begin{aligned} P_j &= P_j^0 = \{(x,y):\ c_j \le \ell(x)\ell(y) < c_{j-1}\} \\ N_j &= N_j^0 = \{(x,y):\ -c_{j-1} < \ell(x)\ell(y) \le -c_j\} \end{aligned}\right\} \quad j=1,2,\ldots,m \tag{11}$$

where c_j's are constants such that $c_0=\infty$, $c_m=0$ and $c_j < c_{j-1}, \forall_j$ and

$$\ell(x) = -f'(x)/f(x). \tag{12}$$

Corollary 1: The threshold parameter $\underline{c}^*$ of an optimum unconstrained 2m-level CC is defined as

$$c_j^* = (w_j^* + w_{j+1}^*)/2\ , \quad j=1,2,\ldots,m-1 \tag{13}$$

and $c_0^*=\infty$ and $c_m^*=0$. The vector $\underline{w}^*$ is given by (9). Theorem 1 can be proved by showing contradiction. Suppose regions P_j (thus N_j), j=1,2,...,m, maximizing the efficacy $E_{\underline{w}^*}$ are not defined as stated in Theorem 1. Then there exists a region δ such that either $\delta \subset P_1$, $\delta \not\subset P_j^0$ but $\delta \subset P_i^0$, or $\delta \not\subset P_j$, but $\delta \subset P_j^0$, $\delta \subset P_i$, for some $i \neq j$ and for an appropriate selection of $\underline{c}$. By deleting δ from P_j and adding it to P_i (or vice versa), we can show an increase in efficacy in both cases. The proof is not difficult but lengthy, so it is not presented in detail. The unconstrained optimum 2m-level CC can be obtained generally by numerical iteration from (9) and (13) with the threshold function g given by

$$g(x,y) = \ell(x)\ell(y)\ . \tag{14}$$

An alternative iterative method is suggested by the proof of Theorem 1. Figure 2 shows the structure of the unconstrained optimum 2m-level CC. From now on we will call it simply the 'optimum CC'.

It is also very simple to establish the following theorem:

Theorem 2: $E_{\underline{w}^*}$ is a non-decreasing function of m.

III. PROPERTIES OF THE OPTIMUM CC

Limiting Efficacy

As a result of Theorem 2, the efficacy for infinitely many levels ought to be the limiting efficacy of an optimum CC. The limiting efficacy for m→∞ means the efficacy for infinitely many and infinitely finer partitions.

In the limiting case, $\ell(x)\ell(y)$ is a constant in any one region for the optimum CC. Since the regions exhaust the data space and ℓ^2 is symmetric about the origin, we can see that

$$\lim_{m\to\infty} E_{\underline{w}^*} = 2 \lim_{m\to\infty} \sum_{j=1}^{m} \frac{\left\{\iint_{P_j} f'f'\right\}^2}{\iint_{P_j} ff}$$

$$= 4 \int_0^\infty \int_0^\infty \ell^2(x)\ell^2(y)f(x)f(y)dxdy$$

$$= I^2(f) \qquad (15)$$

where I(f) is the Fisher's Information for the density f, which is defined as $I(f) = \int_{-\infty}^{\infty} [f'(x)/f(x)]^2 f(x)dx$. In general, the constrained-optimum CC does not achieve the same limiting efficacy. By applying Schwarz's inequality we get

$$\lim_{m\to\infty} E_{\underline{w}^*} = 2 \lim_{m\to\infty} \sum_{j=1}^{m} \frac{\left\{\iint_{P_j} \ell(x)\ (y)f(x)f(y)dxdy\right\}^2}{\iint_{P_j} f(x)f(y)dxdy}$$

$$\leq 2 \lim_{m\to\infty} \sum_{j=1}^{m} \iint_{P_j} \ell^2(x)\ell^2(y)f(x)f(y)dxdy$$

$$= I^2(f)$$

Equality is obtained if in the limiting case $\ell(x)\ell(y)\sqrt{f(x)f(y)} = k\sqrt{f(x)f(y)}$, i.e. $\ell(x)\ell(y)$ is a constant, for $(x,y) \in P_j, \forall_j$. This suggests that the optimum CC is not the only CC which achieves the limiting efficacy $I^2(f)$. An example is considered in the following section. The result allows one to compare the performance of any proposed finite-level CC to its limiting value which is ideally obtainable.

Minimum MSE Fit to $\ell(x)\ell(y)$

From theorem 1 we see that the optimum CC partitions are defined by $P_j^0 = \{(x,y): c_j \leq \ell(x)\ell(y) < c_{j-1}\}$, j=1,...,m, and from (9) and the mean value theorem,

$$w_j^* = \frac{\iint_{P_j} \ell(x)\ell(y)f(x)f(y)dxdy}{\iint_{P_j} f(x)f(y)dxdy} = \ell(\xi_x)\ell(\xi_y)$$

where $(\xi_x,\xi_y)\in P_j$. Thus $c_j \leq w_j^* < c_{j-1}$. This leads one to expect that the optimum CC partitioning approximates the function $\ell(x)\ell(y)$. Let us consider the MSE between $T_p(X_1^i,X_2^i)$ and $\ell(X_1^i)\ell(X_2^i)$. Let $\ell(X_1^i)\ell(X_2^i) = T_c$, with $T_p = T_p(X_1^i,X_2^i)$ being the partition such that $T = \sum_{i=1}^{n} T_p(X_1^i,X_2^i)$ in (1). The MSE ε is given by

$$\varepsilon = E\{T_p - T_c\}^2$$

$$= 2\sum_{j=1}^{m} w_j^2 \iint_{P_j} f_x f_y - 4 \sum_{j=1}^{m} w_j \iint_{P_j} f'_x f'_y + I^2(f) \qquad (16)$$

Let us find the vector $\underline{w}^*$ which minimizes ε for any set of partitions. We get

$$w_j^* = \frac{\iint_{P_j} f'_x f'_y}{\iint_{P_j} f_x f_y} , \quad j=1,2,\ldots,m$$

which is exactly the same result as we obtained in Section II. Substituting w_j^* into (16) yields

$$\varepsilon = I^2(f) - 2 \sum_{j=1}^{m} \frac{\left\{\iint_{P_j} f'_x f'_y\right\}^2}{\iint_{P_j} f_x f_y}$$

$$= I^2(f) - E_{\underline{w}^*} \qquad (17)$$

It is clear that the minimization of ε wrt $\underline{c}$ is the same as the maximization of $E_{\underline{w}^*}$. Thus we may conclude that the optimum CC is the minimum MSE fit to $\ell(x_1^i)\ell(x_2^i)$.

IV. QUANTIZER-CORRELATOR AND GENERALIZED DEAD-ZONE CORRELATOR

As a particular example of a CC, consider a 2q-level quantizer-correlator which quantizes each input into symmetric 2q levels and forms a correlation test based on the quantized data. Consider the test statistic T_{QC} such that

$$T_{QC} = \sum_{j=1}^{m} h(x_1^i)h(x_2^i) \qquad (18)$$

where

$$h(x) = \begin{cases} h_i , & x\varepsilon[a_j, a_{j-1}) \\ -h_j , & x\varepsilon(-a_{j-1}, a_j] \end{cases} , \quad j=1,2,\ldots,q \qquad (19)$$

and $a_0=\infty$, $a_q=0$. Figure 3 shows that the QC partitions the data space into $q(q+1)$ differently weighted regions. It is thus possible to get the optimum set of vectors $\underline{a}$ and $\underline{h}$ which maximizes the efficacy directly from the results obtained in Section II. Applying (8) of Section II, we get the efficacy

$$E_{QC} = 4\left[\frac{\left\{\sum_{j=1}^{q} h_j\left[f(a_j) - f(a_{j-1})\right]\right\}^2}{\sum_{j=1}^{q} h_j^2\left[F(a_{j-1}) - F(a_j)\right]}\right]^2 . \qquad (20)$$

The $\underline{h}^*$ maximizing E_{QC} is given by

$$h_j^* = \frac{f(a_j) - f(a_{j-1})}{F(a_{j-1}) - F(a_j)} , \quad j=1,2,\ldots,q. \tag{21}$$

With $h_j = h_j^*$, (20) becomes

$$E_{QC,\underline{h}^*} = \{2 \sum_{j=1}^{q} [f(a_j) - f(a_{j-1})]^2/[(F(a_{j-1}) - F(a_{j-1})]\}^2 \tag{22}$$

It is also possible to see that E_{QC,h^*} is maximized by the $\underline{a}^*$ which satisfies

$$-\frac{f'(a_j^*)}{f(a_j^*)} = \frac{1}{2} \left\{ \frac{f(a_j) - f(a_{j-1})}{F(a_{j-1}) - F(a_j)} + \frac{f(a_{j+1}) - f(a_j)}{F(a_j) - F(a_{j+1})} \right\}$$

$$= \frac{1}{2} (h_j^* + h_{j+1}^*) \tag{23}$$

Equations (21) and (23) allow us to find the optimum 2q-level QC. It is interesting to note that for infinitely many and infinitely finer quantization-intervals the efficacy $E_{QC,\underline{h}^*}$ approaches $I^2(f)$. In other words the QC is an approximation of the optimum CC for large numbers of levels and its efficacy approaches the limiting value $I^2(f)$ as $q \to \infty$.

The Generalized Dead-zone Coincidence (GDZC) detector [2] is easy to implement (comparable to PCC mentioned earlier) while it maintains relatively good performance. As a direct consequence of (9) and (10), we get for the GDZC (see Fig. 4)

$$w_j^* = \frac{f^2(c_j) - f^2(c_{j-1})}{[1 - F(c_j)]^2 - [1 - F(c_{j-1})]^2} \tag{24}$$

and

$$E_{GDZC,\ \underline{w}^*} = 4 \sum_{j=1}^{m} \frac{[f^2(c_j) - f^2(c_{j-1})]^2}{[1 - F(c_j)]^2 - [1 - F(c_{j-1})]^2} \tag{25}$$

It is easy to see that (25) is maximized by $\underline{c}^*$ satisfying the condition

$$-\frac{f'(c_j^*)}{f(c_j^*)} = \frac{w_j^* + w_{j+1}^*}{2} \frac{[1 - F(c_j^*)]}{f(c_j^*)} , \quad j=1,2,\ldots,m-1 \tag{26}$$

V. NUMERICAL RESULTS

To see how the efficacies of the CC's vary with respect to the number of levels, some numerical results are obtained. Here we consider three different CC's, the optimum CC (unconstrained), the QC and the GDZC detector (both constrained-optimum CC's). Table 1 shows the efficacies obtained for Gaussian noise with zero-mean and unit-variance for different values of m. It is noted that for small m, e.g. m<6, the GDZC performs better than the QC which is a good approximation of the optimum CC for large m.

REFERENCES

1. H. Ekre, "Polarity Coincidence Correlation Detection of a Weak Noise Source," IRE Trans. Inf. Theory, Vol. IT-9, pp. 18-23, 1963.
2. S. A. Kassam and J. B. Thomas, "Improved Nonparametric Coincidence Detectors," J. Franklin Inst., Vol. 303, pp. 75-84, 1977.
3. S. A. Kassam and J. B. Thomas, "Dead-zone Limiter: An Application of Conditional Tests in Nonparametric Detection," J. Acoust. Soc. Am., Vol. 60, No. 4, pp. 857-862, 1976.
4. S. A. Kassam and J. B. Thomas, "Generalizations of the Sign Detector Based on Conditional Tests," IEEE Trans. Comm., Vol. COM-24, pp. 481-487, 1976.
5. D.A.S. Fraser, Nonparametric Methods in Statistics (John Wiley & Sons, New York, 1957).
6. J. B. Thomas, "Nonparametric Detection," Proc. IEEE, Vol. 58, pp. 623-631, 1970.
7. J. Capon, "On the Asymptotic Efficiency of Locally Optimum Detectors," IEEE Trans. Inf. Theory, IT-7, pp. 67-71, 1961.

APPENDIX
(Derivation of (8))

Since the mean of T under H_0 is zero and the observations are assumed independent, we get

$$\mathrm{Var}\{T|H_0\} = n \sum_{j=1}^{m} [w_j^2\, P\{(x_1,x_2)\varepsilon P_j|H_0\} + (-w_j)^2 P\{(x_1,x_2)\varepsilon N_j|H_0\}]$$

$$= n \sum_{j=1}^{m} w_j^2 [\iint_{P_j} f(x)f(y)dxdy + \iint_{N_j} f(x)f(y)dxdy] \tag{A1}$$

Similarly

$$E_\theta\{T|H_1\} = n\, E_s[\sum_{j=1}^{m} w_j\{\iint_{P_j} f(x-\theta s)f(y-\theta s)dxdy - \iint_{N_j} f(x-\theta s)f(y-\theta s)dxdy\}], \tag{A2}$$

$$\frac{d}{d\theta^2} E_\theta\{T|H_1\}\Big|_{\theta=0} = \frac{1}{2\theta}\frac{d}{d\theta} E_\theta\{T|H_1\}\Big|_{\theta=0}$$

$$= n \sum_{j=1}^{m} w_j\{\iint_{P_j} f_x' f_y' - \iint_{N_j} f_x' f_y'\}\ . \tag{A3}$$

From (7) we obtain

$$E_{2m} = \frac{\{\sum_{j=1}^{m} w_j[\iint_{P_j} f_x' f_y' - \iint_{N_j} f_x' f_y']\}^2}{\sum_{j=1}^{m} w_j^2[\iint_{P_j} f_x f_y + \iint_{N_j} f_x f_y]} \tag{A4}$$

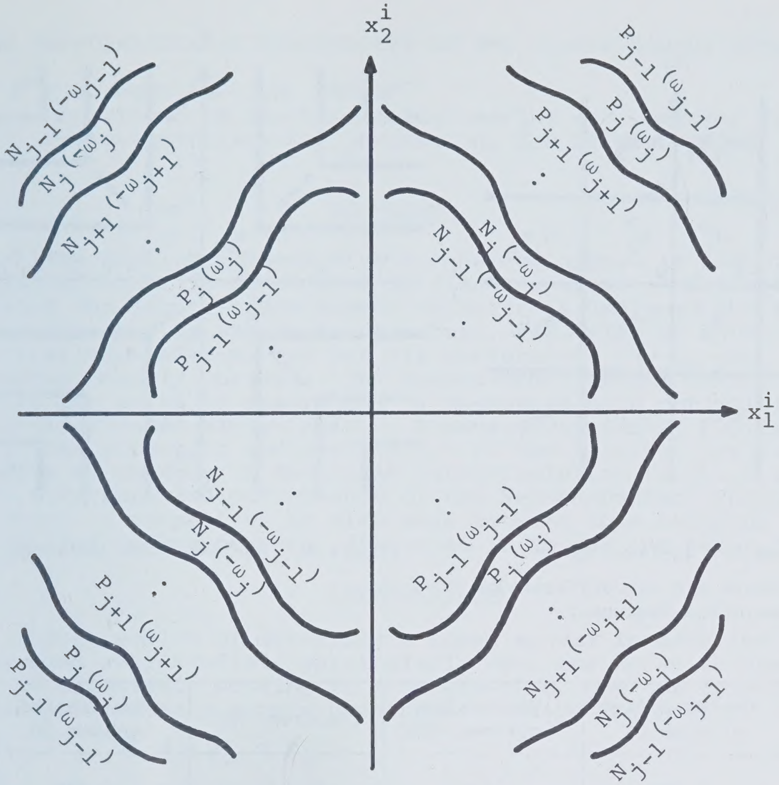

x_2^i
x_1^i
$N_{j-1}(-\omega_{j-1})$
$N_j(-\omega_j)$
$N_{j+1}(-\omega_{j+1})$
$P_{j-1}(\omega_{j-1})$
$P_j(\omega_j)$
$P_{j+1}(\omega_{j+1})$
$P_j(\omega_j)$
$P_{j-1}(\omega_{j-1})$
$N_j(-\omega_j)$
$N_{j-1}(-\omega_{j-1})$
$N_{j-1}(-\omega_{j-1})$
$N_j(-\omega_j)$
$P_{j-1}(\omega_{j-1})$
$P_j(\omega_j)$
$P_{j+1}(\omega_{j+1})$
$P_j(\omega_j)$
$P_{j-1}(\omega_{j-1})$
$N_{j+1}(-\omega_{j+1})$
$N_j(-\omega_j)$
$N_{j-1}(-\omega_{j-1})$

Figure 1. Two-dimensional quantization for 2m-level CC

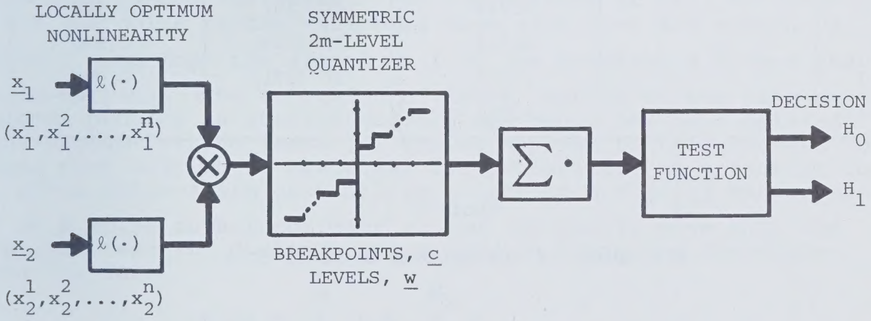

LOCALLY OPTIMUM NONLINEARITY
$\underline{x}_1$
$\ell(\cdot)$
$(x_1^1, x_1^2, \ldots, x_1^n)$
$\underline{x}_2$
$\ell(\cdot)$
$(x_2^1, x_2^2, \ldots, x_2^n)$
SYMMETRIC 2m-LEVEL QUANTIZER
BREAKPOINTS, $\underline{c}$
LEVELS, $\underline{w}$
$\Sigma\cdot$
TEST FUNCTION
DECISION
H_0
H_1

Figure 2. Structure of the unconstrained optimum 2m-level CC (For nonparametric testing a conditioning loop may be added [4].)

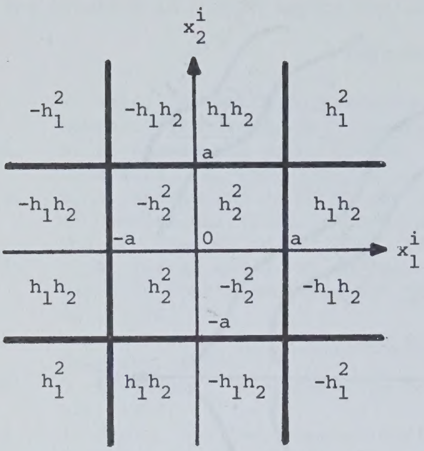

x_2^i
$-h_1^2$
$-h_1h_2$
h_1h_2
h_1^2
a
$-h_1h_2$
$-h_2^2$
h_2^2
h_1h_2
-a
0
a
x_1^i
h_1h_2
h_2^2
$-h_2^2$
$-h_1h_2$
-a
h_1^2
h_1h_2
$-h_1h_2$
$-h_1^2$

Figure 3. 2q-level QC:q=2

(Note there are six differently weighted regions.)

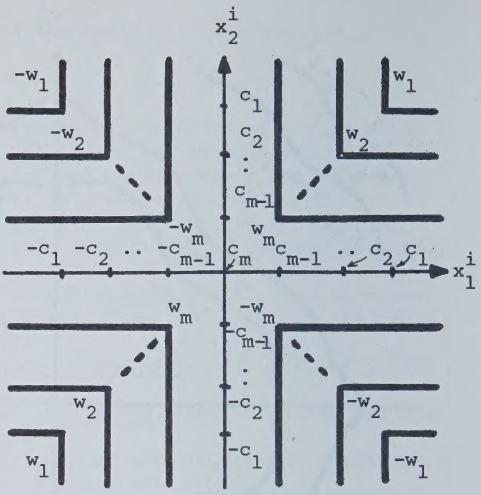

x_2^i
$-w_1$
c_1
w_1
$-w_2$
c_2
w_2
c_{m-1}
$-w_m$
w_m
$-c_1$
$-c_2$
$-c_{m-1}$
c_m
c_{m-1}
c_2
c_1
x_1^i
w_m
$-w_m$
$-c_{m-1}$
w_2
$-c_2$
$-w_2$
w_1
$-c_1$
$-w_1$

Figure 4. 2m-level GDZC detector

m	Unconstrained optimum CC	Constrained optimum GDZC	q(corres. m)	Constrained optimum QC
2	0.790	0.739	2 (3)	0.779
3	0.893	0.830	3 (6)	0.887
4	0.935	0.867	4 (10)	0.932
5	0.956	0.886	5 (15)	0.955
6	0.969	0.897	6 (21)	0.968
10		0.915	10 (55)	0.988
15		0.921	15 (120)	0.994
21		0.923	18 (171)	0.996
28		0.924		

Table 1.

Efficacies of the optimum 2m-level CC's

THE PERFORMANCE AND SENSITIVITY OF TWO FINITE-MEMORY DETECTORS

M.D. WOOD AND J.B. THOMAS
Department of Electrical Engineering and Computer Science
Princeton University, Princeton, New Jersey, 08540.

ABSTRACT

The problem of detecting a constant signal in additive Gaussian noise is considered in discrete-time. A detector called the second-order Markov detector is designed and analyzed. This detector is simple in structure, requiring no inversion of a covariance matrix, and yet its performance is, in many instances, nearly optimal. The second-order Markov detector contains two units of memory and is Neyman-Pearson optimal in second-order Gauss-Markov noise. A sensitivity study indicates that the structure and performance of the detector are less sensitive to changes in the noise autocorrelation function than are the structure and performance of the Neyman-Pearson optimal detector. A comparison is also made between this detector and the so-called Markov detector discussed in an earlier paper [4].

I. INTRODUCTION

The problem of detecting a known signal in additive noise has been well-studied, particularly when the noise is assumed to be Gaussian. Working in discrete-time, one can formulate this problem as a simple hypothesis testing problem:

$$\begin{aligned} H_0&: x_i = n_i \\ H_1&: x_i = n_i + s_i \end{aligned} \qquad i = 1,2,\ldots,N$$

where the N observations are denoted by $\{x_i\}_{i=1}^{N}$, the $\{s_i\}_{i=1}^{N}$ represent a known signal, and the noise samples $\{n_i\}_{i=1}^{N}$ are realizations of a sequence $\{N_i\}_{i=1}^{N}$ of identically distributed Gaussian random variables. For convenience it will be assumed that the noise random variables have zero mean and common variance σ^2 and that the signal $\{s_i\}$ is the constant s rather than time-varying. The extension of these results to the time-varying signal problem is straightforward and hence has been omitted. The Neyman-Pearson lemma [1] can be applied to this problem to find that test which maximizes the probability of detection for a given false-alarm probability. Letting $R = [r_{ij}]$ denote the N by N noise covariance matrix, one can easily show that the optimum test for distinguishing between these two hypotheses has the form

$$t(\underline{x}) = \sum_{i=1}^{N} h_i x_i \underset{H_0}{\overset{H_1}{\gtrless}} T$$

where the coefficients $\{h_i\}_{i=1}^{N}$ satisfy the well-known [2] matched filter equation, $R\underline{h} = \underline{s}$, and the threshold T is chosen to give the desired probability of false-alarm. (The under-bar will be used throughout this paper to denote vector quantities.) Unfortunately, solving the matched filter equation for large N is in general a difficult and time-consuming computational task. It is also true that the filter coefficients $\{h_i\}$ can be quite sensitive to perturbations in the noise covariance matrix R. For these reasons, the Neyman-Pearson optimal detector is perhaps less practical than one would like.

One is lead by the above considerations to search for detectors which are simple in structure but which retain a reasonably high level of performance. The Neyman-Pearson optimal white-noise detector is certainly simple in structure, consisting only of a linearity $g_{ind}(x) = (2sx-s^2)/2\sigma^2$, a summer, and a threshold comparator. However, its performance in correlated noise is in many cases less than satisfactory. Intuitively, one would expect that the performance of this white-noise detector in colored Gaussian noise could be improved by incorporating some memory into the system. Adding a delay of one time unit and a second linearity $h(\cdot)$ to the optimal white-noise linearity $g_{ind}(\cdot)$ results in a detector which has been called the Markov detector. Whereas the white-noise detector is memoryless, this modified detector structure contains one unit of memory and was studied in a previous paper [4]. In that paper the linearity $h(\cdot)$ was chosen so as to maximize the power of the detector for a given probability of false-alarm. In the present paper we wish to investigate the performance and sensitivity of a detector having two units of memory. This detector is obtained by adding two time delays and two linearities, $h_1(\cdot)$ and $h_2(\cdot)$, to the optimal white-noise detector. For reasons which will become apparent shortly, this 2-memory detector will be called the second-order Markov detector and is depicted in Fig. 1.

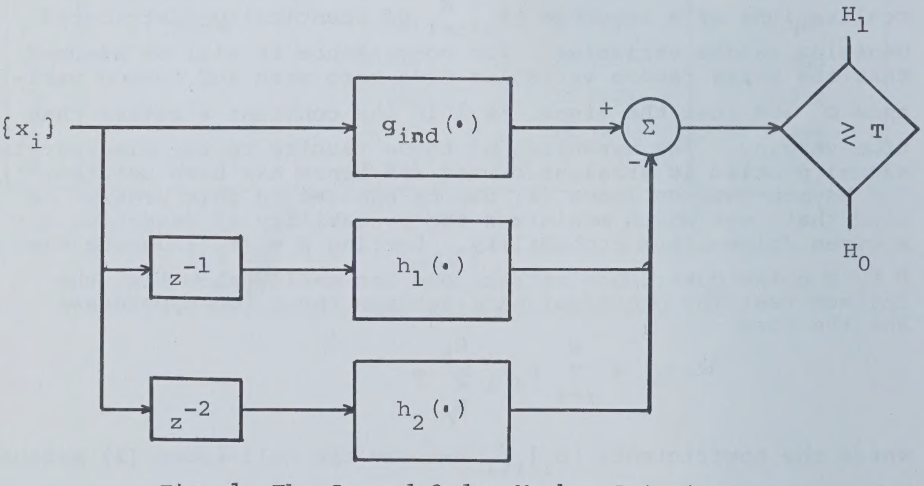


Fig. 1 The Second-Order Markov Detector

II. OPTIMALITY PROPERTIES OF THE SECOND-ORDER MARKOV DETECTOR

In this section it will be shown that the linearities $h_1(\cdot)$ and $h_2(\cdot)$ can be chosen so that the second-order Markov detector of Fig. 1 is Neyman-Pearson optimal when the noise random variables are Gaussian and possess the second-order Markov property [3]. It will also be shown that in first-order Gauss-Markov noise, this detector reduces to the Markov detector, and in white Gaussian noise, it reduces to the optimal white-noise detector.

With the $\{N_i\}_{i=1}^{N}$ assumed to be a second-order Markov sequence, we can write [3]

$$\begin{aligned} f_N(n_1,n_2,\ldots,n_N) &= f_2(n_1,n_2) \prod_{i=3}^{N} f_c(n_i|n_{i-2},n_{i-1}) \\ &= f_2(n_1,n_2) \prod_{i=3}^{N} \frac{f_3(n_{i-2},n_{i-1},n_i)}{f_2(n_{i-2},n_{i-1})} \end{aligned} \tag{1}$$

where $f_k(\cdot,\cdot,\ldots,\cdot)$ represents the kth-order probability density function of the noise random variables and $f_c(\cdot|\cdot,\cdot)$ represents the conditional probability density function of the first variable given the latter two. The likelihood ratio $L(\underline{x})$ can then be written

$$\begin{aligned} L(\underline{x}) &\triangleq p_1(\underline{x})/p_0(\underline{x}) \\ &= f_N(x_1-s,x_2-s,\ldots,x_N-s)/f_N(x_1,x_2,\ldots,x_N) \qquad (2) \\ &= \frac{f_2(x_1-s,x_2-s)}{f_2(x_1,x_2)} \prod_{i=3}^{N} \frac{f_3(x_{i-2}-s,x_{i-1}-s,x_i-s)}{f_3(x_{i-2},x_{i-1},x_i)} \cdot \frac{f_2(x_{i-2},x_{i-1})}{f_2(x_{i-2}-s,x_{i-1}-s)} \end{aligned}$$

On applying the Gaussian assumption, taking logarithms, and simplifying, Eq. (2) becomes

$$\begin{aligned} \ell(\underline{x}) &\triangleq \ell n L(\underline{x}) \\ &= Cs \sum_{i=3}^{N} [(x_{i-2}-s/2)(\rho-1)+(x_{i-1}-s/2)(2\rho-\bar{\rho}-1)+(x_i-s/2)(\rho-1)] \\ &\quad - \frac{s}{\sigma^2(1+\rho)} \sum_{i=4}^{N} [(x_{i-2}-s/2)+(x_{i-1}-s/2)] \end{aligned} \tag{3}$$

where $E[N_i] = 0$, $E[N_i^2] = \sigma^2$, $E[N_iN_{i+1}]/\sigma^2 = \rho$, $E[N_iN_{i+2}]/\sigma^2 = \bar{\rho}$, and $C = 1/(2\rho^2-\bar{\rho}-1)\sigma^2$. Considerable manipulation of Eq. (3) yields the result

$$\begin{aligned} \ell(\underline{x}) &= \frac{s}{\sigma^2(1+\rho)} \sum_{i=1}^{N} (x_i-s/2) - \frac{\rho s C(\bar{\rho}-1)}{1+\rho} \sum_{i=3}^{N} (x_{i-1}-s/2) \\ &\quad - \frac{2sC(\bar{\rho}-\rho^2)}{1+\rho} \sum_{i=5}^{N} (x_{i-2}-s/2) - \frac{sC(\bar{\rho}-\rho^2)}{1+\rho}(x_1+x_2+x_{N-1}+x_N) . \end{aligned} \tag{4}$$

According to the Neyman-Pearson lemma [1], the test which maximizes the detection probability β for a given false-alarm probability α is a test of the form

$$\ell(\underline{x}) \underset{H_0}{\overset{H_1}{\gtrless}} T \tag{5}$$

where the threshold T is determined through the relation $\alpha = P_0[\ell(\underline{x}) > T]$. Multiplying both sides of Eq. (5) by the non-negative quantity $(1 + \rho)$ and dropping the extraneous term*, we obtain an equivalent test $\hat{\ell}(\underline{x}) \underset{H_0}{\overset{H_1}{\gtrless}} \hat{T}$ where

$$\hat{\ell}(\underline{x}) = \frac{s}{\sigma^2}\sum_{i=1}^{N}(x_i - s/2) - sC\rho(\bar{\rho}-1)\sum_{i=3}^{N}(x_{i-1}-s/2) - 2sC(\bar{\rho}-\rho^2)\sum_{i=5}^{N}(x_{i-2}-s/2) \tag{6}$$

and

$$\hat{T} = T(1 + \rho).$$

Recognizing that $g_{ind}(x_i) = (s/\sigma^2)(x_i - s/2)$ is the optimal white-noise linearity and letting $h_1(x_{i-1}) = sC\rho(\bar{\rho}-1)(x_{i-1}-s/2)$ and $h_2(x_{i-2}) = 2sC(\bar{\rho}-\rho^2)(x_{i-2}-s/2)$, it should be clear that this optimal test can be implemented using the detector structure in Fig. 1. That is, the second-order Markov detector is Neyman-Pearson optimal when the noise is Gaussian and second-order Markov. Notice that the linearities $g_{ind}(\cdot)$, $h_1(\cdot)$, and $h_2(\cdot)$ have different slopes but they all have the same zero-crossing point, namely $x = s/2$. If the $\{N_i\}_{i=1}^{N}$ constitute a first-order Markov sequence, Eq. (1) reduces to

$$f_N(n_1, n_2, \ldots, n_N) = f(n_1)\prod_{i=2}^{N} f_c(n_i|n_{i-1})$$

and Eq. (3) simplifies to

$$\ell(\underline{x}) \triangleq \ell nL(\underline{x}) = \frac{s}{\sigma^2(1+\rho)}\sum_{i=1}^{N}(x_i - s/2) - \frac{\rho s}{\sigma^2(1+\rho)}\sum_{i=3}^{N}(x_{i-1} - s/2).$$

Multiplying this test statistic and its corresponding threshold by the factor $(1 + \rho)$ results in the Markov detector described in [4]. When the noise random variables are statistically independent, the quantities ρ and $\bar{\rho}$ are identically zero and Eq. (4) reduces easily to $\ell(\underline{x}) = \sum_{i=1}^{N} g_{ind}(x_i)$, the optimal white-noise detector.

---

*The last term on the right-hand side of Eq. (4) is a result of the fact that the first two samples and the last two samples, x_1, x_2, x_{N-1}, and x_N, do not have a full set of "neighbors." It can be shown that, except for very small values of N, dropping this term has very little effect on either the performance or the sensitivity of the detector.

III. EVALUATION OF THE PERFORMANCE OF THE DETECTOR

From Eq. (6) it is clear that the test statistic $\hat{\ell}(\underline{x})$ of the second-order Markov detector has the form

$$\hat{\ell}(\underline{x}) = \sum_{i=1}^{N} g_{ind}(x_i) - a \sum_{i=3}^{N} (x_{i-1}-s/2) - b \sum_{i=5}^{N} (x_{i-2}-s/2) \qquad (7)$$

where a and b are the slopes of the linearities $h_1(\cdot)$ and $h_2(\cdot)$, respectively. Under the hypothesis H_0, the quantity $\hat{\ell}(\underline{x})$ is a Gaussian random variable with mean $-\mu_{ab}$ and variance σ^2_{ab} where $\mu_{ab} = Ns^2/2\sigma^2 - (N-2)as/2 - (N-4)bs/2$,

$$\sigma^2_{ab} = \frac{s^2}{\sigma^4} \sum_{i=1}^{N} \sum_{j=1}^{N} r_{ij} + a^2 \sum_3^N \sum_3^N r_{ij} + b^2 \sum_5^N \sum_5^N r_{ij} - \frac{2as}{\sigma^2} \sum_1^N \sum_3^N r_{i,j-1}$$

$$- \frac{2bs}{\sigma^2} \sum_1^N \sum_5^N r_{i,j-2} + 2ab \sum_3^N \sum_5^N r_{i-1,j-2} ,$$

and

$$r_{ij} = E[N_i N_j] .$$

Under the alternative H_1, the test statistic $\hat{\ell}(\underline{x})$ has mean μ_{ab} and the same variance σ^2_{ab}. The power β_{ab} of this detector, as defined by the relation $\beta_{ab} = P_1[\hat{\ell}(\underline{x}) > \hat{T}]$, is computed to be

$$\beta_{ab} = 1 - \Phi[\Phi^{-1}(1-\alpha) - 2\mu_{ab}/\sigma_{ab}] \qquad (8)$$

where $\Phi(\cdot)$ is the unit normal cumulative distribution function, $\Phi^{-1}(\cdot)$ is its inverse, and α is the desired false-alarm probability.

If the noise is known to be first- or second-order Markov, or if the noise is white, then the optimal values for the slope parameters a and b (optimal in the Neyman-Pearson sense) are known and were given in the previous section. If the noise has none of these properties, then the parameters a and b can be chosen so that the power of the detector, as given by Eq. (8), is maximized. Using an optimization technique such as the well-known Fletcher-Powell algorithm [5], optimal values a* and b* are easily found with the aid of a digital computer.

Equation (8), which gives the detection probability as a function of α, represents the receiver operating characteristic of the second-order Markov detector operating in correlated Gaussian noise. We will now compute the operating characteristics of the white-noise detector, the Markov detector, and the Neyman-Pearson optimal detector in order that we might compare the performance of these four detectors.

As mentioned previously, the Neyman-Pearson optimal detector for a constant signal in additive white Gaussian noise has the form

$$d(\underline{x}) \triangleq \sum_{1}^{N} g_{ind}(x_i) = (s/\sigma^2) \sum_{i=1}^{N} (x_i - s/2) \underset{H_0}{\overset{H_1}{\gtrless}} D$$

where again the threshold D is chosen so as to attain the desired probability of false-alarm. The power β_{ind} of this detector, operating in a correlated noise environment, is found to be

$$\beta_{ind} = 1 - \Phi[\Phi^{-1}(1-\alpha) - 2\mu_{ind}/\sigma_{ind}]$$

where $\mu_{ind} = Ns^2/2\sigma^2$ is the mean of the test statistic $d(\underline{x})$ under H_1, the quantity $-\mu_{ind}$ is its mean under H_0, and $\sigma^2_{ind} = (s^2/\sigma^4) \sum_{1}^{N} \sum_{1}^{N} r_{ij}$ is its variance under either hypothesis.

The performance of the Markov detector was evaluated in [4]. It was shown there that the Markov detector is Neyman-Pearson optimal in Gauss-Markov noise and has a receiver operating characteristic given by

$$\beta_M = 1 - \Phi[\Phi^{-1}(1-\alpha) - 2\mu_M/\sigma_M]$$

where

$$\mu_M = Ns^2/2\sigma^2 - (N-2)ms/2 \ ,$$

$$\sigma_M^2 = \frac{s^2}{\sigma^4} \sum_{i=1}^{N} \sum_{j=1}^{N} r_{ij} + m^2 \sum_{2}^{N-1} \sum_{2}^{N-1} r_{ij} - \frac{2ms}{\sigma^2} \sum_{1}^{N} \sum_{2}^{N-1} r_{ij} \ ,$$

and

$$m^* = \frac{(s/\sigma^2)\left[(N-2) \sum_{1}^{N} \sum_{1}^{N} r_{ij} - N \sum_{1}^{N} \sum_{2}^{N-1} r_{ij}\right]}{(N-2) \sum_{1}^{N} \sum_{2}^{N-1} r_{ij} - N \sum_{2}^{N-1} \sum_{2}^{N-1} r_{ij}}$$

is the optimal value of the parameter m denoting the slope of the linearity h(•).

The Neyman-Pearson optimal test for detecting a constant signal in correlated Gaussian noise, as mentioned in Section I, can be written

$$t(\underline{x}) = \sum_{i=1}^{N} \sum_{j=1}^{N} q_{ij} x_i s$$

where x_i, $i = 1,2,\ldots,N$ are the observations, s is the amplitude of the signal to be detected, and q_{ij} are the elements of the inverse of the noise covariance matrix $[q_{ij}] = [r_{ij}]^{-1}$. Assuming the input noise samples to have zero mean and unit variance, Zubakov [6] has shown that

$$E_0[t(\underline{x})] = 0 \ , \ \mathrm{Var}_0[t(\underline{x})] = s^2 \sum_{1}^{N} \sum_{1}^{N} q_{ij} \triangleq K \ ,$$

and

$$E_1[t(\underline{x})] = K \ , \ \mathrm{Var}_1[t(\underline{x})] = K \ .$$

The power of this detector, which is the maximum attainable by any detector with false-alarm probability α, is easily computed to be

$$\beta_{opt} = 1 - \Phi[\Phi^{-1}(1-\alpha) - \sqrt{K}] \ .$$

The power ratios β_{opt}/β_{ind}, β_{ab}/β_{ind}, and β_M/β_{ind} have been computed for a wide variety of noise autocorrelation functions. As discussed in [4], the Markov detector performs substantially better in colored Gaussian noise than does the white-noise detector and, in some instances, its performance is nearly optimal. As to be expected, the second-order Markov detector performs even better. Some typical examples are presented in Figs. 2 and 3, where the various power ratios are plotted as functions of N. It has been observed that all three ratios are unimodal whenever the underlying noise autocorrelation function is unimodal, and that, regardless of the noise autocorrelation, each ratio asymptotically approaches unity. If the noise autocorrelation function is not unimodal, the behavior of the different power ratios depends to a rather large extent on how the autocorrelation function is sampled.

IV. SENSITIVITY STUDY OF THE SECOND-ORDER MARKOV DETECTOR

Until now it has been assumed that all the quantities needed to design the second-order Markov detector were known exactly. In many practical situations, these quantities are not known exactly and hence it is of considerable interest to investigate the effects of imperfect knowledge of these quantities on the performance and structure of the detector. In this section the sensitivity of the second-order Markov detector to changes in the underlying noise autocorrelation function will be examined.

In general, the optimal slope parameters a* and b* depend not only on the noise autocorrelation function $R_n(\tau)$ but also on N, the number of observations. One would hope that small changes in N or small perturbations in $R_n(\tau)$ would produce only small changes in a* and b*. In reference [4], this was found to be true for the single slope parameter m* contained in the Markov detector. Unfortunately (but not unexpectedly), numerical studies indicate that the parameters a* and b* are significantly less robust with respect to $R_n(\tau)$ and N than was m*. Whereas, for a fixed value of N, the quantity m* was seen to depend more on the time duration of $R_n(\tau)$ than on the actual function itself, a* and b* depend critically on the particular functional form of $R_n(\tau)$. This fact is indicated by the table in Fig. 4. The table also indicates that for a fixed autocorrelation function, a* and b* vary more as functions of N than m* did.

More quantitative results concerning the sensitivity of a* and b* are found in Fig. 5. For two different one-parameter autocorrelation functions,

$$R_n(\tau) = \begin{cases} 1 - 2|\tau|/T \ , & |\tau| \le T/2 \\ & \text{, otherwise} \end{cases}$$

and

$$R_n(\tau) = \text{sech}(\gamma\tau) \ ,$$

the parameters T and γ, respectively, were perturbed from a nominal value by a specified amount and the resulting changes in a* and b* were computed. As indicated by Fig. 5, the structure of the second-order Markov detector is in general slightly more sensitive to scale changes in $R_n(\tau)$ than was the Markov detector of reference [4].

A similar study was made in [4] of the sensitivity of the Neyman-Pearson optimal detector. Recall that the Neyman-Pearson optimal constant-signal detector can be written in the form $t(\underline{x}) = \sum_{i=1}^{N} h_i x_i$ where the filter coefficients satisfy the matched filter equation, $R\underline{h} = \underline{s}$. In that study, several different autocorrelation functions were perturbed by a specified scale factor and the resulting changes in the h_i's were computed. For our two example $R_n(\tau)$'s, the results are tabulated in Fig. 6. In general, only the minimum change in the $\{h_i\}$ is comparable to the corresponding changes in a* and b*. The average change in the $\{h_i\}$ is considerably larger and the maximum change is orders of magnitude larger. It appears from these studies that the relatively simple structure of the second-order Markov detector is significantly less sensitive than the Neyman-Pearson optimal detector to changes in the noise autocorrelation function.

A more important issue concerning changes in the noise autocorrelation function is the effect such changes have on the performance of the detector. Numerical studies have been conducted in an attempt to obtain a quantitative measure of how much the power of the second-order Markov detector changes when the noise autocorrelation function is perturbed. In these studies the second-order Markov detector was designed for a given autocorrelation function and the power of the detector operating in that noise environment was computed. Keeping the detector structure exactly the same, the noise autocorrelation function was changed by a scale factor and the resulting change in power, $\Delta\beta_{ab}$, was calculated. For the two examples, $R_n(\tau)$ = triangle and $R_n(\tau) = \mathrm{sech}(\gamma\tau)$, the results appear in Fig. 5. It is of interest to note that the changes $\Delta\beta_{ab}$ reported here are almost identical in value to the changes $\Delta\beta_M$ in the power of the Markov detector observed when an identical study was performed on that detector[4]. When the same procedure is applied to the Neyman-Pearson optimal detector, the resulting changes in power $\Delta\beta_{opt}$ are much larger than either $\Delta\beta_{ab}$ or $\Delta\beta_M$. (See Fig. 6.) Hence the performance of either of the two Markov detectors is less affected by changes in the noise autocorrelation function than is the performance of the optimal detector.

ACKNOWLEDGEMENTS

The authors are grateful for the support of the National Science Foundation under Grant ENG-76-19808 and the U.S. Army Research Office under Grant DAAG29-75-G-0192.

REFERENCES

1. S.D. Silvey, Statistical Inference, New York: Wiley,1975,p.98.
2. C.W. Helstrom, Statistical Theory of Signal Detection, Oxford: Pergamon Press, 1968, Chapter IV.
3. J.L. Doob, Stochastic Processes, New York: Wiley, 1953, p.89.
4. M.D. Wood and J.B. Thomas, "A Nearly Optimal Detector for Constant Signals in Gaussian Noise," submitted to IEEE Trans. on Communications.
5. R. Fletcher and M. Powell, "A Rapid Descent Method for Minimization," Computer Journal, Vol. 6, No. 2, 1963, pp.163-168.
6. V.D. Zubakov, "Optimum Detection in the Presence of Correlated Noise," Rad. Eng. and Elect., Vol. III, Dec. 1958, pp.43-56.

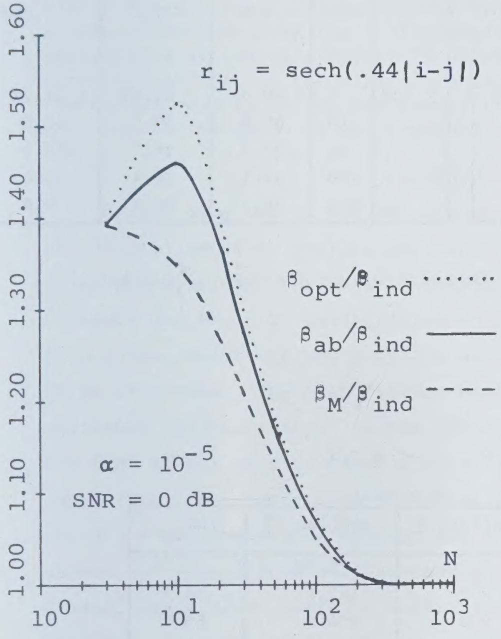


Fig. 2 Typical Power Ratios

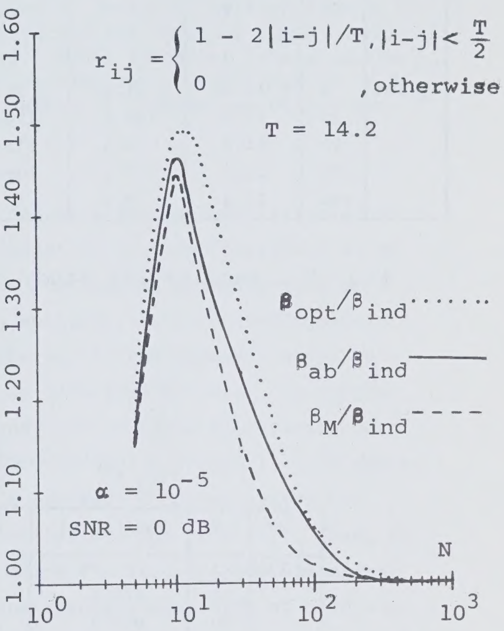


Fig. 3 Typical Power Ratios

$R_n(\tau)$ / N	$e^{-\gamma\|\tau\|}$		shot noise*		triangle		sech$(\gamma\tau)$		$e^{-\gamma^2\tau^2}$		$\frac{\sin(\gamma\tau)}{\gamma\tau}$		$e^{-\gamma\|\tau\|}\cos(\Omega\tau)$	
	a*	b*	a*	b*	a*	b*	a*	b*	a*	b*	a*	b*	a*	b*
5	.75	0.	.95	.001	1.00	.00	3.93	-6.95	2.39	-3.08	4.79	-9.36	.86	-.003
10	.75	0.	.85	.02	.62	.28	1.88	-1.02	1.49	-.65	2.58	-1.80	.85	-.02
50	.75	0.	1.04	-.23	1.11	-.29	1.55	-.65	1.37	-.50	1.77	-.82	.92	-.13
100	.75	0.	1.04	-.24	1.11	-.29	1.54	-.64	1.36	-.50	1.73	-.79	.93	-.15
1000	.75	0.	1.05	-.24	1.12	-.30	1.53	-.63	1.36	-.50	1.74	-.80	.94	-.16

$\alpha = 10^{-5}$

SNR = 0 dB

$\Delta\tau_1$ = time duration = 7.00

$r_{ij} = R_n(|i-j|)$

$$*R_n(\tau) = \begin{cases} (1-|\frac{\tau}{T}|)^2(1+|\frac{\tau}{2T}|) & , |\tau| < T \\ 0 & , |\tau| > T \end{cases}$$

Fig. 4 Typical Values of a* and b*

$$r_{ij} = \begin{cases} 1 - 2|i-j|/T \,, & |i-j| < T/2 \\ 0 & \text{, otherwise} \end{cases}$$

$$T_o = 14.2$$

ΔT	N	Δa^*	Δb^*	$\Delta\beta_{ab}$
± 10%	5	± 0.0%	± 0.0%	± 8.0%
	10	28.1	85.9	10.1
	50	1.8	.6	23.9
	100	1.9	1.1	19.4
	500	1.9	1.4	.004
± 50%	5	± 11.4%	± 11.4%	± 73.6%
	10	53.7	145.5	114.8
	50	11.9	6.4	211.2
	100	12.5	7.6	105.5
	500	12.8	9.0	.2

$$r_{ij} = \operatorname{sech}(\gamma \cdot |i-j|)$$

$$\gamma_o = .44$$

$\Delta\gamma$	N	Δa^*	Δb^*	$\Delta\beta_{ab}$
± 10%	5	± 9.8%	± 16.6%	± 5.0%
	10	5.6	9.8	12.0
	50	4.3	7.2	23.2
	100	4.2	7.2	19.2
	500	4.2	7.2	.004
± 50%	5	± 49.3%	± 87.6%	± 24.7%
	10	30.0	57.0	60.8
	50	21.2	36.7	115.8
	100	21.1	36.4	85.0
	500	20.7	36.3	2.0

Fig. 5 Sensitivity Study of Second-Order Markov Detector

$$r_{ij} = \begin{cases} 1 - 2|i-j|/T \,, & |i-j| \le T/2 \\ 0 & \text{, otherwise} \end{cases}$$

ΔT	N	$\max_i\{\lvert\Delta h_i\rvert\}$	$\min_i\{\lvert\Delta h_i\rvert\}$	$\operatorname{ave}_i\{\lvert\Delta h_i\rvert\}$	$\lvert\Delta\beta_{opt}\rvert$
± 10%	3	83.3%	1.7%	28.9%	9.5%
	10	413.6	4.7	123.2	30.7
	50	807.0	2.5	256.6	69.6
	100	360.2	2.5	84.4	55.3
± 50%	50	2148.5	2.1	232.0	94.5

$$r_{ij} = \operatorname{sech}(\gamma \cdot |i-j|)$$

$\Delta\gamma$	N	$\max_i\{\lvert\Delta h_i\rvert\}$	$\min_i\{\lvert\Delta h_i\rvert\}$	$\operatorname{ave}_i\{\lvert\Delta h_i\rvert\}$	$\lvert\Delta\beta_{opt}\rvert$
± 10%	3	23.6%	14.4%	17.5%	6.1%
	10	62.1	17.9	38.6	33.1
	50	260.9	1.9	46.2	69.7
	100	260.9	1.9	28.1	54.8
± 50%	50	413.9	3.8	81.9	758.5

Fig. 6 Sensitivity Study of N.-P. Optimal Detector

ON UPPER AND LOWER BOUNDS OF ERROR PROBABILITY OF A FINITE-PRECISION DIGITALLY IMPLEMENTED RECEIVER*

S. REISENFELD and K. YAO
Department of System Science
University of California
Los Angeles, CA 90024

ABSTRACT

In this paper, we provide a new approach to the evaluation of error probability of an all digitally implemented receiver. Specifically, we consider a binary signal detection problem over an AWGN channel, where the receiver utilizes a fixed-point arithmetic, a finite-level quantizer, and a finite dynamic-range scaling device followed by a digital matched filter. A relationship between the system scaling-factor and an upper bound on the probability of system overflow is obtained. Upper and lower bounds on the error probability of the digital receiver are derived as functions of E_b/N_o, the number of bits of the processor, the number of samples per binary symbol, and the matched filter structure.

I. INTRODUCTION

Recent progress in microelectronics has made the implementation of an all-digital receiver attractive. Thus, the study of these receivers is of great practical and theoretical interest. Degradation in error-rate performance due to A/D quantization noise, multiplicative roundoff quantization noise, and processor overflow are inherent in the implementation of these receivers. The quantization noise is minimized by using the largest available dynamic range. On the other hand, random signals operating near the full system dynamic range limitation have a finite probability of reaching saturation. Saturation is undesirable because it causes system nonlinearity and system degradation for a correlation type receiver. Thus, we choose the appropriate scaling factor to allow the largest possible signal dynamic range with a certain tolerable constrained probability of overflow. With this scaling factor, we are interested in obtaining the error probability of this all-digital receiver. The effect of A/D quantization noise and internal filter roundoff noise due to multiplication is considered. The error probability is a function of E_b/N_o, the number of bits of the processor, the number of samples per binary symbol, and the matched-filter structure. In general, the exact evaluation of this P_e expression is analytically intractable. All commonly encountered digital filtering error analyses are based essentially upon the mean-square error criterion and are not particularly relevant to a digital receiver analysis. In this paper, we use the moment-space bounding technique to obtain tight upper and lower

---

* This work was supported by the Electronics Program of the Office of Naval Research.

bounds on P_e. Some specific numerical examples are considered.

Figure 1 shows a block diagram of the system. The signal set consists of binary, antipodal, baseband waveforms. The received waveform over the observed interval $[0,T]$ is given by,

$$r(t) = \begin{cases} -S(t) + n(t) , & \text{under } H_0 \\ S(t) + n(t) , & \text{under } H_1 \end{cases} ,$$

where $S(t)$ has energy E_b and $n(t)$ is a Gaussian process with zero mean and has two-sided noise density $N_o/2$. The transfer function of the low-pass filter $H(f)$ has unity magnitude for $|f| \le B$ and zero elsewhere. The filtered waveform $\hat{r}(t)$ is sampled at a rate of $f_s = 2B$ samples/second. The amplifier has a voltage scaling factor of K. The scaling factor is chosen to insure that the quantizer and the digital signal processor have a low probability of overflow. It is well known that, for an infinite resolution receiver, the optimum signal detector is a correlation receiver or a matched filter. For a finite precision system operated in a condition such that the probability of overflow is small, the system is essentially linear and thus matched filtering will yield close to optimum performance.

The quantizer output and all processor registers are constrained to be $b+1$ binary numbers where b bits carry magnitude information, and one bit carries sign information. The quantizer function $Q(r)$ is analytically described by

$$Q(r) = \begin{cases} m2^{-b} , & \text{for } (m-\tfrac{1}{2})2^{-b} \le r < (m+\tfrac{1}{2})2^{-b} \\ & \qquad \text{where } -(2^b-1) \le m \le 2^b-1 \\ 1-2^{-b} , & \text{for } (1-\tfrac{1}{2}2^{-b}) \le r \\ -(1-2^{-b}) , & \text{for } r < -(1-\tfrac{1}{2}2^{-b}) \end{cases} .$$

The digital filter shown in Figure 2 is a one-pole recursive filter using fixed-point arithmetic. The filter impulse response is given by,

$$h_k = \alpha^k , \qquad k \ge 0 , \tag{1}$$

where $\alpha \in [0,1]$. In each T second interval, L samples are processed for detection, where $L = 2BT$. From Eq. (1) and implementation of the digital filter as a matched filter, the sampled transmitted waveform is,

$$S(\tfrac{i}{L} T) = A\alpha^{L-i} , \quad 1 \le i \le L . \tag{2}$$

The sequence at the sampler output is given by,

$$r_i = \begin{cases} -S(\frac{i}{L} T) + n_i , & H_0 , \\ & & 1 \le i \le L \\ S(\frac{i}{L} T) + n_i , & H_1 , \end{cases}$$

where $\{n_i\}$ is a zero-mean uncorrelated Gaussian random sequence of variance $N_o B$.

Figure 2 shows the equivalent noise model of the digital receiver. The A/D converter noise sequence $\{\varepsilon_{AD,i}\}$ and the multiplicative filter round-off noise sequence $\{\varepsilon_{F,i}\}$ are assumed to be mutually independent, zero-mean, uniformly distributed, independent random sequences with variances σ^2_{AD} and σ^2_F , respectively. Then $\sigma^2_{AD} = \sigma^2_F = (2^{-2b})/12$.

The problem to be considered here is the evaluation of P_e as a function of E_b/N_o , b , L , and α . A basic assumption in our analysis is that the probability of processor overflow is controlled to be sufficiently small so that error-rate degradation due to overflow is negligible and will be neglected.

II. THE DYNAMIC RANGE CONSTRAINT

The quantizer input has a finite dynamic range and therefore the band-limited, sampled channel observations must be adjusted in amplitude by the scaling factor K so that appropriate signal dynamic range is achieved. If K is too small, excessive P_e degradation due to quantization noise will result and, if K is too large, the effect will be significant P_e degradation due to A/D and filter overflows.

One method of setting the dynamic range of the stochastic signal X is to constrain it to be contained with high probability in the interval $[-\theta(X,\gamma),\theta(X,\gamma)]$ where

$$\theta(X,\gamma) \overset{\Delta}{=} |E[X/H_i]| + \gamma \, \mathrm{Var}^{1/2}[X/H_i] \tag{3}$$

and γ is a positive parameter.

It can be shown that, for any given ε greater than zero, there exists a sufficiently large γ so that the probability of overflow satisfies

$$\Pr\{|X| > \theta(X,\gamma)\} < \varepsilon \quad . \tag{4}$$

The output of the matched filter at time k is given by

$$y_k = (-1)^{m+1}\left[\sum_{i=1}^{k} h_{k-i}\, KS(\tfrac{i}{L} T)\right] + N_{c,k} + N_{AD,k} + N_{F,k}$$

$$1 \le k \le L$$

$$= (-1)^{m+1} X_k + N_{c,k} + N_{AD,k} + N_{F,k} , \tag{5}$$

where $m \in \{0,1\}$ is the index for hypothesis H_m , $N_{c,k}$ is the Gaussian channel noise, $N_{AD,k}$ is the A/D quantization noise, and $N_{F,k}$ is the multiplicative roundoff noise. It can be shown that ,

$$\theta(y_i,\gamma) \le \theta(y_L,\gamma) , \qquad 1 \le i \le L , \tag{6}$$

and

$$\theta(Kr_i,\gamma) \le \theta(y_L,\gamma) , \qquad 1 \le i \le L . \tag{7}$$

These two results show that an upper bound on the probability of overflow of the matched filter at the end of the detection interval provides an upper bound on overflow probability for the matched filter output and quantizer input for all sample times. Thus, a single γ sets an upper bound on the probability of overflow for the digital receiver for all sample times.

In order to operate the receiver without appreciable overflow or excessive quantization noises, we choose the smallest γ satisfying Eq. (4) and the largest K consistent with this γ , b , α , A , L , and E_b/N_o . The relationships between this maximum allowable K and these other system parameters are given by

$$K = \frac{\lambda}{A\beta\mu}\left\{1 - \gamma\left[\frac{\eta\beta\mu}{\lambda^2} + \frac{1}{2\rho}\right]^{1/2}\right\} \tag{8}$$

where

$$\rho = E_b/N_o$$

$$\lambda = 1 - \tfrac{1}{2}2^{-b}$$

$$\eta = \frac{2^{-2b}}{6}$$

$$\mu = 1 - \frac{\gamma^2}{2\rho}$$

$$\beta = \frac{1 - \alpha^{2L}}{1-\alpha^2}$$

III. RECEIVER ERROR PROBABILITY

The output of the matched filter at the end of the decision instant is given by Eq.(5), evaluated at $k = L$. The probability of decision error, P_e, is given by

$$\begin{aligned} P_e &= \frac{1}{2}\Pr\{y_L > 0/H_0\} + \frac{1}{2}\Pr\{y_L < 0/H_1\} \\ &= \Pr\{N_{c,L} > X_L - N\} \\ &= \Pr\{N_{c,L} > X_L + N\} \\ &= E_N\{\mathrm{erfc}[\sqrt{2\rho}\,(1 + \frac{N}{AK\beta})]\} \end{aligned} \tag{9}$$

where $N = N_{AD,L} + N_{F,L}$, and $\mathrm{erfc}(\cdot)$ is the complement of the probability distribution of the zero mean and unit variance Gaussian random variable.

Direct evaluation of the expectation in the P_e expression of Eq.(9) is computationally unattractive, since the explicit evaluation of the probability density of the random variable N is difficult. Thus, we obtain upper and lower bounds on the P_e expression of Eq.(9) by using only moment information of the random variable N, which is relatively easy to obtain. One possible method of obtaining these bounds on P_e is the moment-space technique [1]-[3].

A relatively simple type of bounds may be obtained by using information only in the variance of the random variable N. Specifically, the variance of N can be expressed as $\sigma_N^2 = \beta((2^{-2b})/6)$. The explicit form of these bounds may be found in references [1]-[2]. The computation of σ_N^2 may be made using only the assumption that $\{\varepsilon_{AD,i}\}$ and $\{\varepsilon_{F,i}\}$ are each uncorrelated and mutually uncorrelated. The uncorrelated assumption of quantization noise sequences has been shown to be quite appropriate in many signal processing applications [4]-[5].

A family of tighter bounds, using higher order moments of the random variable N, can be obtained [3]. The computation of the higher order moments of the random variable N can be readily obtained by using the recursive formula given in [6]. A sufficient condition for the evaluation of these higher order moments is the mutual independence of all the components of N.

IV. NUMERICAL RESULTS

In order to illustrate the concepts discussed in Sections II and III, some numerical results are presented here. The error probability expression given by Eq.(9) depends upon the following independent variables: the

$\mathrm{SNR(dB)} = 10 \log_{10}(E_b/N_o)$; the matched filter decay factor α ; the signal dynamic range factor γ ; the number of samples L per binary decision, and the number of bits $(b+1)$ per register.

In Table I, we present upper bounds $P_e{}^U$ and lower bounds $P_e{}^L$ of the digital receiver which are parameterized by some typical values of b and SNR for $\alpha = 0.5$, $\gamma = 4$, and $L = 8$. Let $M_i = E_N[1 + N/AK\beta]^i$. For $(b+1) = 4$ and 5 , M_2 , M_4 , M_6 , and M_8 were used to compute bounds on P_e . For $(b+1) = 6$, 8, 10, and 12, M_2 and M_4 were used to compute bounds on P_e.

The effect of increasing the number of moments used on the tightness of the P_e bounds is shown in Table II. The parameter values are fixed at $\alpha = 0.5$, $L = 8$, $\gamma = 4.0$, and $b = 5$. Bounds obtained from 1, 2, 3, and 4 moments are shown.

Table III shows the variation in P_e as α is changed.

V. CONCLUSION

A new approach, the moment-space bounding technique, to the evaluation of the theoretical P_e performance of an all digitally implemented receiver has been presented. A relationship between receiver amplitude scaling factor and the probability of overflow in the receiver was given. The error-rate computation included the effects of internal receiver quantization noise but excluded the effect of overflow. These results are applicable to situations where the probability of overflow is sufficiently small so that P_e contributions from overflow are negligible.

Table I shows a P_e vs. binary wordlength trade-off. At SNR = 10 dB, for the parameters shown, the performance begins to degrade significantly at a 5-bit wordlength. At SNR = 8 dB and 6 dB , the performance begins to degrade significantly at a 4-bit wordlength.

As more moments are used, the P_e bounds become tighter. As the SNR decreases, the bounds also become tighter. These effects can be seen in Table II.

Table III shows that P_e increases monotonically with α . This indicates that receivers implemented with the smallest possible α are desirable. Since the peak instantaneous transmitted power requirement increases with decreasing α , the smallest α consistent with peak transmitted power capability should be selected.

Generalizations of the analysis to include the effect of overflow on P_e and to generalize to more complex matched-filter structures should be possible. This paper is a preliminary analytical characterization of the interaction of system parameters and performance in an all-digital receiver.

REFERENCES

1. Matthews, J.W.: Sharp error bounds for intersymbol interference, *IEEE Trans.Inform. Theory,* IT-19 (1973) 440-447.
2. Yao, K., and Tobin, R.M.: Moment space upper and lower bounds for digital systems with intersymbol interference, *IEEE Trans.Inform.Theory,* IT-22 (1976) 65-74.
3. Yao, K. and Biglieri, E.: Moment inequalities for error probabilities in digital communication systems, *Proc.Nat.Telecomm.Conf., Dec. 1977.*
4. Bennett, W.R.: Spectra of quantized signals, *Bell Syst.Tech.J.,* 27 (1948).
5. Oppenheim, A.V., and Weinstein, C.J.: Effects of finite register length in digital filters and Fast Fourier Transform,*Proc.IEEE,* 60 (1972) 957-976.
6. Prabhu, V.K.: Some considerations of error bounds in digital systems,*Bell Syst.Tech.J.,* 50:10 (1971) 3127-3151.

TABLE I: P_e vs. Receiver Binary Wordlength (b+1)

SNR (dB)		*Probability of Bit Error* (P_e)						
		(b+1)=4	(b+1)=5	(b+1)=6	(b+1)=8	(b+1)=10	(b+1)=12	(b+1)=∞
6	P_e^U:	6.87×10^{-3}	2.90×10^{-3}	2.46×10^{-3}	2.40×10^{-3}	2.39×10^{-3}	2.39×10^{-3}	2.39×10^{-3}
	P_e^L:	4.54×10^{-3}	2.86×10^{-3}	2.37×10^{-3}	2.39×10^{-3}	2.39×10^{-3}	2.39×10^{-3}	
8	P_e^U:	1.63×10^{-3}	2.79×10^{-4}	2.00×10^{-4}	1.92×10^{-4}	1.91×10^{-4}	1.91×10^{-4}	1.91×10^{-4}
	P_e^L:	5.63×10^{-4}	2.64×10^{-4}	1.90×10^{-4}	1.92×10^{-4}	1.91×10^{-4}	1.91×10^{-4}	
10	P_e^U:	2.94×10^{-4}	8.80×10^{-6}	3.98×10^{-6}	3.92×10^{-6}	3.88×10^{-6}	3.87×10^{-6}	3.87×10^{-6}
	P_e^L:	2.30×10^{-5}	7.04×10^{-6}	3.84×10^{-6}	3.90×10^{-6}	3.87×10^{-6}	3.87×10^{-6}	

Parameters: L = 8 , $\alpha = 0.5$, $\gamma = 4.0$

TABLE II: Tightness of P_e Bounds vs. Number of Moments Used

SNR (dB)		P_e			
		$\{M_2\}$	$\{M_2,M_4\}$	$\{M_2,M_4,M_6\}$	$\{M_2,M_4,M_6,M_8\}$
6	P_e^U:	4.79×10^{-3}	2.57×10^{-3}	2.52×10^{-3}	2.50×10^{-3}
	P_e^L:	2.37×10^{-3}	2.46×10^{-3}	2.49×10^{-3}	2.50×10^{-3}
8	P_e^U:	5.76×10^{-4}	2.24×10^{-4}	2.13×10^{-4}	2.08×10^{-4}
	P_e^L:	1.90×10^{-4}	2.00×10^{-4}	2.07×10^{-4}	2.08×10^{-4}
10	P_e^U:	2.27×10^{-5}	5.58×10^{-6}	4.94×10^{-6}	4.59×10^{-6}
	P_e^L:	3.84×10^{-6}	4.19×10^{-6}	4.53×10^{-6}	4.56×10^{-6}

Parameters: $L=8$, $\alpha=0.5$, $\gamma=4.0$, $b=5$

TABLE III: P_e vs. α

SNR (dB)		P_e			
		$\alpha = 0$	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 0.9$
6	P_e^U:	2.394×10^{-3}	2.394×10^{-3}	2.396×10^{-3}	2.419×10^{-3}
	P_e^L:	2.393×10^{-3}	2.393×10^{-3}	2.394×10^{-3}	2.403×10^{-3}
8	P_e^U:	1.917×10^{-4}	1.917×10^{-4}	1.921×10^{-4}	1.961×10^{-4}
	P_e^L:	1.916×10^{-4}	1.916×10^{-4}	1.918×10^{-4}	1.930×10^{-4}
10	P_e^U:	3.905×10^{-6}	3.906×10^{-6}	3.921×10^{-6}	4.122×10^{-6}
	P_e^L:	3.899×10^{-6}	3.899×10^{-6}	3.905×10^{-6}	3.946×10^{-6}

Parameters: $L=8$, $\gamma=4.0$, $b=7$

Moments Used: $\{M_2,M_4\}$

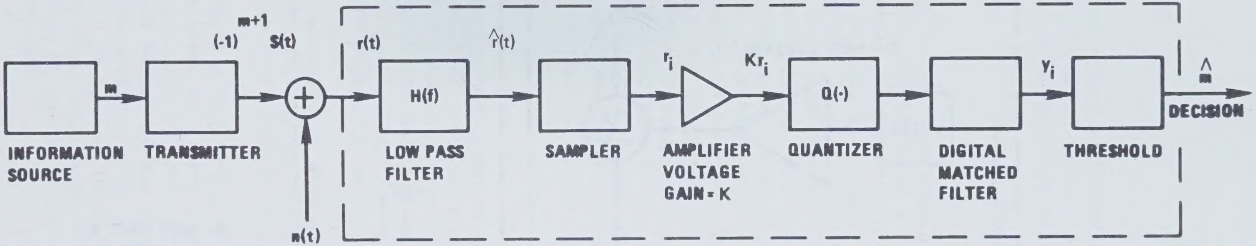

INFORMATION SOURCE
m
TRANSMITTER
$(-1)^{m+1}$
S(t)
n(t)
r(t)
H(f)
LOW PASS FILTER
$\hat{r}(t)$
SAMPLER
r_i
AMPLIFIER VOLTAGE GAIN = K
Kr_i
Q(·)
QUANTIZER
DIGITAL MATCHED FILTER
y_i
THRESHOLD
$\hat{m}$
DECISION

Figure 1. The block diagram of the communication system implemented with an all digital receiver.

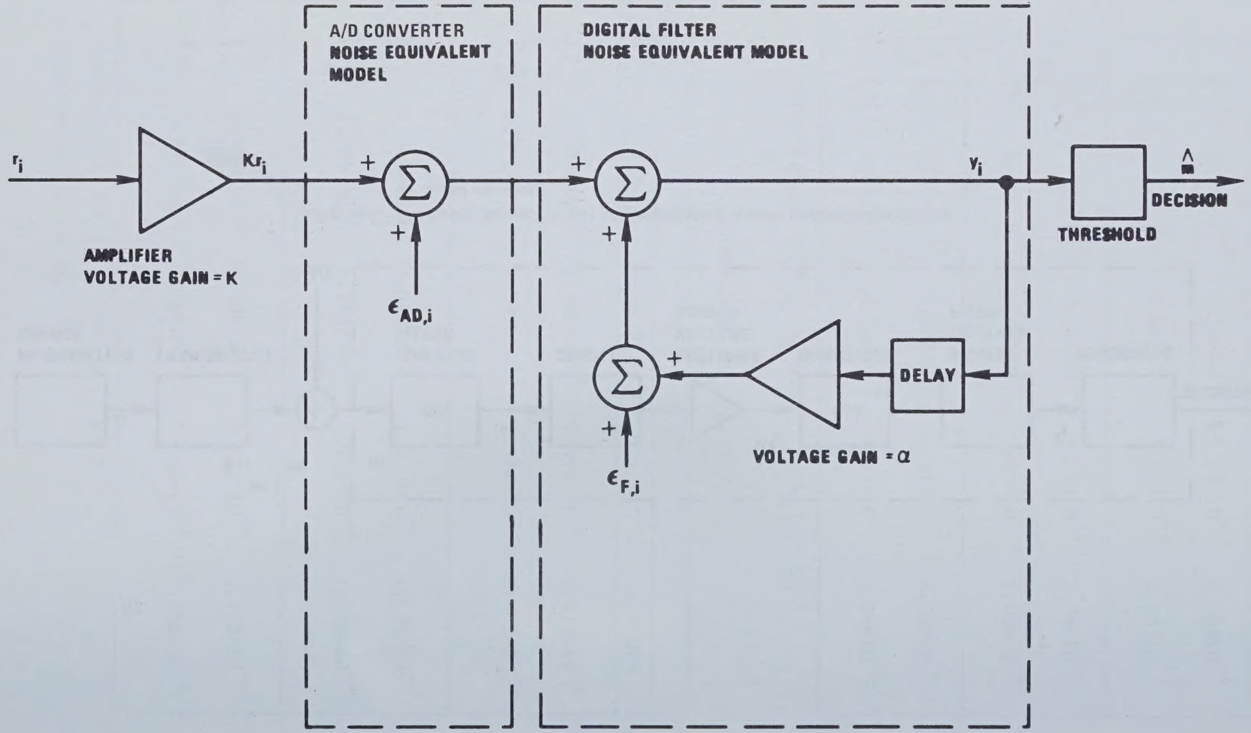

A/D CONVERTER NOISE EQUIVALENT MODEL
DIGITAL FILTER NOISE EQUIVALENT MODEL
r_i
Kr_i
AMPLIFIER VOLTAGE GAIN = K
+
+
$\epsilon_{AD,i}$
+
+
+
+
$\epsilon_{F,i}$
y_i
DELAY
VOLTAGE GAIN = α
THRESHOLD
$\hat{m}$
DECISION

Figure 2. The equivalent noise model of the digital receiver.

AN ALGORITHM FOR FINDING HAMILTONIAN CIRCUITS IN CERTAIN GRAPHS

ROBERT E. BIXBY
Northwestern University

and

DA-LUN WANG
University of Kentucky

Abstract

A direct algorithmic procedure is described for computing Hamiltonian circuits in graphs which satisfy a condition due to Chvátal. For a graph with p vertices this procedure works in $O(p^3)$ operations.

1. Introduction.

V. Chvátal [3] has given the best possible Dirac type [4] sufficient conditions for the existence of a Hamiltonian circuit in an undirected linear graph. We present an $O(p^3)$ algorithm for constructing a Hamiltonian circuit in a graph with p vertices, when the existence of such a circuit is guaranteed by Chvátal's condition. Independently of our work Bondy and Chvátal [2] have also discovered an $O(p^3)$ algorithm which accomplishes the same end. Indeed their procedure is much simpler than ours. However, our interest in the present work is not so much a function of its simplicity as its possible use in the development of heuristic procedures for finding long (perhaps Hamiltonian) paths in graphs which may not satisfy Chvátal's condition. For this purpose our algorithm may be more appropriate since it is a more or less direct procedure for constructing paths. The Bondy-Chvátal algorithm, on the other hand, involves a clever preprocessing step which really produces nothing until the existence of a Hamiltonian circuit is guaranteed.

2. Definitions.

We define a (finite undirected) graph G to be a finite set E(G) of edges and a finite set V(G) of vertices together with a relation of incidence that associates with each edge two vertices called its ends. An edge is said to join its ends. A graph in which each edge has distinct ends (i.e., no loops), and no two edges have exactly the same ends (i.e., no multiple edges) is called a simple graph.

In this paper we deal exclusively with simple graphs, since even if loops and multiple edges were allowed, they would play no role; consequently, the world ''simple'' will be henceforth omitted.

Let G be a graph. When $A = \{x,y\} \in E(G)$ we say that x and y are adjacent, and that A is incident to x (and y). A path P in G is a sequence of vertices $P = (x_1,\ldots,x_n)$, $n \geq 1$, of G such that $x_2,\ldots,x_n$ are distinct and consecutive terms of the sequence are adjacent. x_1 and x_n are called the end vertices of P. P^{-1}, the reverse of P, denote the path$(x_n,\ldots,x_1)$. If $x_1 = x_n$ we say P is a circuit. A circuit in G which includes every vertex is called Hamiltonian. G itself is called Hamiltonian if it has a Hamiltonian circuit. Clearly, a Hamiltonian graph is connected.

Assume G has p vertices. For $x \in V(G)$ the degree of x, denoted $d(x)$, is the number of edges of G incident to x. Now let $x_1,\ldots,x_p$ be an enumeration of $V(G)$ chosen so that $d(x_i) \leq d(x_j)$ if $i < j$. Put $d_i = d(x_i)$, then the sequence $d_1,\ldots,d_p$ is called the degree sequence of G.

3. Chvátal's Theorem.

The basis of our work is

Theorem 3.1 (Chvátal [3]). Let G be a graph with p 3 vertices and degree sequence $d_1 \leq \ldots \leq d_p$. Suppose that

(1) $d_k \leq k < p/2 \Rightarrow d_{p-k} \geq p-k$. Then G is Hamiltonian.

As noted in the introduction, Theorem 3.1 is the best possible result of a certain type. In particular, Chvátal has shown that if a graph G on p vertices fails to satisfy (1), then there is a graph $H = K_k + (\overline{K}_k \cup K_{p-2k})$ on p vertices whose i-th degree is larger than that of G, and H is not Hamiltonian.

Our interest in Theorem 3.1 stems from an interest in combinatorial procedures for finding Hamiltonian circuits in general graphs. Of course it is well-known that this problem is very difficult. It is therefore reasonable to look at heuristics, and it seems to us that a good theoretical way to search for the right heuristics is to take a nice sufficiency theorem, such as Theorem 3.1, and provide a direct algorithmic proof in the hope that the steps employed by the algorithm may have applications to general graphs.

4. The Algorithm.

We establish some notation to be used in describing the algorithm. Let G be a graph with p 3 vertices, and suppose G satisfies (1). Denote $E(G)$ by E and $V(G)$ by V. Let $v_1,\ldots,v_p$ be an enumeration of V so that $d(v_1)$ ... $d(v_p)$. For each x V define $\sigma(x)$ to be the first integer such that $x = v_{\sigma(x)}^p$.

In the description of the algorithm that follows P denotes a path. We write the generic P as $(x_1,\ldots,x_n)$. The algorithm will begin with P equal to a single vertex and terminate with P equal to a Hamiltonian circuit.

In the latter part of the algorithm we introduce paths H and L. These paths will be denoted generically by $L = (y_1,\ldots,y\ )$ and $H = (z_1,\ldots,z_h)$

ALGORITHM

Step 1. Set $P \leftarrow (v_p)$.

Step 2. If $\exists$ vertex y off P s.t. $\{y,x_1\}$ or $\{x_n,y\} \in E$, then go to Step 3. Otherwise set $P \leftarrow (y, x_1,\ldots,x_n)$ or $P \leftarrow (x_1,\ldots,x_n, y)$, as the case may be, and repeat Step 2.

Step 3. If $d(x_1) + d(x_n) < n$ go to Step 5. Otherwise let i, $2 \leq i \leq n$, be s.t. $\{x_1, x_i\}, \{x_{i-1}, x_n\} \in E$.

Set $P \leftarrow (x_1, x_2, \ldots, x_{i-1}, x_n, x_{n-1}, \ldots, x_i, x_1)$.

Step 4. If $n = p$ STOP. Otherwise find i, $1 \leq i \leq n-1$, and a vertex y off P st. $\{x_i, y\} \in E$.

Set $p \leftarrow (y, x_i, x_{i-1}, \ldots, x_1, x_{n-1}, \ldots, x_{i+1})$, and go to Step 2.

Step 5. If $d(x_1) > d(x_n)$, set $P \leftarrow P^{-1}$.

Step 6. If $\exists$ i, $3 \leq i \leq n$, s.t. $\{x_1, x_i\} \in E$ and (a) $d(x_{i-1}) > d(x_1)$ or (b) $\exists$ vertex y off P s.t. $\{x_{i-1}, y\} \in E$, then set $P \leftarrow (x_{i-1}, x_{i-2}, \ldots, x_1, x_i, x_{i+1}, \ldots, x_n)$ and go to Step 2.

Step 7. If $n < p$ go to Step 9.

Step 8. If $\exists$ i, $1 \leq i \leq n-2$, s.t. $\{x_n, x_i\} \in E$ and $d(x_{i+1}) > d(x_n)$, then set $P \leftarrow (x_{i+1}, x_{i+2}, \ldots, x_n, x_i, x_{i-1}, \ldots, x_1)$ and go to Step 3.

Step 9. If $\exists$ i, $3 \leq i \leq n$, s.t. $d(x_i) + d(x_1) \geq p$ and $d(x_{i-1}) > d(x_1)$ then go to Step 10. Otherwise go to Step 11.

Step 10. If $\{x_1, x_n\} \in E$ set $P \leftarrow (x_1, \ldots, x_n, x_1)$ and go to Step 4. If E j, $3 \leq j \leq i-1$, s.t. $\{x_1, x_j\}, \{x_{j-1}, x_i\} \in E$, set $P \leftarrow (x_n, x_{n-1}, \ldots, x_i, x_{j-1}, x_{j-2}, \ldots, x_1, x_j, x_{j+1}, \ldots, x_{i-1})$ and go to Step 2.

If $\exists$ j, $i + 2 \leq j \leq n-1$, s.t. $\{x_1, x_j\}, \{x_i, x_{j+1}\} \in E$, set $P \leftarrow (x_{i-1}, x_{i-2}, \ldots, x_1, x_j, x_{j-1}, \ldots, x_i, x_{j+1}, x_{j+2}, \ldots, x_n)$ and go to Step 2.

Step 11. Let $s' \geq k = d(x_1)$ be as large as possible s.t. for $r = k, \ldots, s'$ we have v_{p-r} on P and $d(x_{i-1}) \leq r$ where $v_{p-r} = x_i$. Set $s \leftarrow s' + 1$. If v_{p-s} is off P let $1 \leq i \leq n-1$ be s.t. $\{x_i, v_{p-s}\} \in E$ and $\sigma(x_{i+1}) > p - s$. Otherwise let $2 \leq i \leq n - 1$ be s.t. $v_{p-s} = x_{i+1}$. In either case set $L \leftarrow (x_1, \ldots, x_i)$ and $H \leftarrow (x_{i+1}, \ldots, x_n)$.

Step 12. If $d(z_1) < p - k$ go to Step 13. (a) If $\exists\ i, 1 \leq i \leq \ell-1$, s.t. $\{z_1,y_i\}, \{y_1,y_{i+1}\} \in E$, then set $P \leftarrow (z_h,z_{h-1},\ldots,z_1,y_i,y_{i-1},\ldots,y_1, y_{i+1},y_{i+2},\ldots,y_\ell)$ and go to Step 2. (b) If $\exists\ i, 1 \leq i \leq h-1$, s.t. $\{y_1,z_i\}, \{z_1,z_{i+1}\} \in E$ set $P \leftarrow (y_\ell,y_{\ell-1},\ldots,y_1, z_i,z_{i-1},\ldots, z_1,z_{i+1},z_{i+2},\ldots,z_h)$ and go to Step 2.

(c) Set $P \leftarrow (y_\ell,\ldots,y_1,z_h,\ldots,z_1)$ and go to Step 2.

Step 13. (a) If $\exists\ i, 1 \leq i \leq \ell-1$, s.t. $\{z_1,y_i\} \in E$ and $\sigma(y_{i+1}) > \sigma(z_1)$, set $L \leftarrow (y_1,y_2,\ldots,y_i,z_1,z_2,\ldots,z_h)$, $H \leftarrow (y_{i+1}, y_{i+2},\ldots,y_\ell)$ and go to Step 12.

(b) If $\exists\ i, 1 \leq i \leq h-1$, s.t. $\sigma(z_i) > \sigma(z_1)$ and

and go to Step 12. (c) Set $H \leftarrow H^{-1}$ and go to Step 12.

$\{z_1,z_{i+1}\} \in E$, set $H \leftarrow (z_i,z_{i-1},\ldots,z_1,z_{i+1},z_{i+2},\ldots,z_h)$

5. Proof That the Algorithm Works

In this section, we show that the algorithm of section 4 works. We will only list the following Lemmas, and the details of the proof will appear in the Journal of Mathematical Programming.

Lemma 5.1. Suppose Step 2 has just been completed and $d(x_1) + d(x_n) \geq n$. Then $\exists\ i, 2 \leq i \leq n$, s.t. $\{x_1,x_i\}, \{x_{i-1},x_n\} \in E$. (See Step 3.)

Lemma 5.2. G is connected. In particular, any non-Hamiltonian circuit in G has a vertex joined to a vertex outside itself. (See Step 4.)

Lemma 5.3. Assume Step 6 has just failed, $\{x_1,x_n\} \notin E$ and $\exists\ i, 3 \leq i \leq n$, s.t. $d(x_i) + d(x_1) \geq p$ and $d(x_{i-1}) > d(x_1)$. Then either $\exists\ j, 3 \leq j \leq i-1$, s.t. $\{x_1,x_j\}, \{x_{j-1}, x_i\} \in E$ or $\exists\ j, i+1 \leq j \leq n-1$, s.t. $\{x_1,x_j\}, \{x_i,x_{j+1}\} \in E$. (See Step 10.)

Lemma 5.4. Set $k = d(x_1)$ and assume we have just begun execution of Step 11. Then we have the following conclusions.

(a) $v_p,\ldots,v_{p-k}$ are on P and $d_{p-k} \geq p-k$.

(b) s' as defined in Step 11 exists and $s' < \frac{p}{2} - 1$.

(c) If we set $s = s' + 1$, then $d(v_{p-i}) \geq p-i$ for $k \leq i \leq s$, and if v_{p-s} is not on P, then $\exists\, i,\ 2 \leq i \leq n-1$, s.t. $\{x_i, v_{p-s}\} \in E$ and $\sigma(x_{i+1}) > p-s$.

Lemma 5.5. Suppose we have begun execution of Step 12 and we find $d(z_1) \geq p-k$. Then if Steps 12(a) and 12(b) fail we must have $\{y_1, z_h\} \in E$.

Lemma 5.6. Suppose Steps 13(a) and (b) have just failed. Then $\sigma(z_h) > \sigma(z_1)$.

Lemma 5.7. Suppose Step 11 has just been executed. Put $r = s-k+1$. Then after at most r applications each of Steps 12 and 13 we will return to Step 2. In addition, we will have that either

(a) $d(P)$ has been increased by at least r, or

(b) applying Step 2 results in increasing the length of P.

The following is our main result.

Theorem 5.8. Let G be a graph on $p \geq 3$ vertices satisfying (1). Then steps 1 through 13 find a Hamiltonian circuit in G in $O(p^3)$ operations.

We close this section with a discussion of some notions relevant to Steps 11,12 and 13. Define an *r-edge rearrangement* of a path P to be a path Q with the same vertices as P and no more than r edges not in P. For example, Figure 3 illustrates all of the *proper* 2-edge rearrangements,

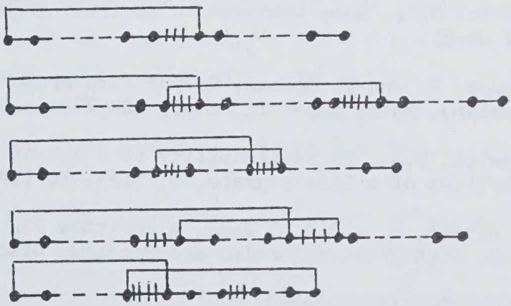


Fig. 3

Note that Steps 6 and 8 of our algorithm involve 1-edge rearrangements, while Step 10 involves 2-edge rearrangements. It would be nice from the point of view of constructing heuristics, if every step in the algorithm involved only r-edge rearrangements with r independent of p. However, as Theorem 5.9 shows, no such procedure can subsume the one described in Steps 11, 12 and 13.

Let us denote as a C-rearrangement a rearrangement which, apart from vertex degree considerations, is achievable by Steps 11, 12 and 13. Then we have

Theorem 5.9. For any integer $\ell > 0$ $\exists$ C-rearrangement which is not achievable as a sequence of r-step rearrangements for any $r < \ell$.

Bibliography

1. Bondy, J.A., Properties of graphs with constraints on degrees, Studia Sci. Math. Hungar. 4 (1969) 473-475.

2. Bondy, J.A. and V. Chvátal, A method in graph theory, Discrete Math., 16 (1976) 111-135.

3. Chvátal, V., On Hamilton's ideals, J. Combin. Theory 12 (1972) 163-168.

4. Dirac, G.A., Some theorems on abstract graphs, Proc. London Math. Soc. 2 (1952) 68-81.

5. Erdős, P. and T. Gallai, Graphs with prescribed degrees on vertices (Hungarian), Math. Lapok 11 (1960) 264-274.

6. Hakimi, S.L., On realizability of a set of integers as degrees of the vertices of a linear graph, J. SIAM, 10 (1962) 496-506.

7. Kleitman, D. and D.L. Wang, Algorithms for constructing graphs and digraphs with given valencies and factors, Discrete Math. 6(1973) 79-88.

8. Pósa, L., (to appear).

FAST ALGORITHMS FOR FINDING MINIMUM SPANNING TREES IN K DIMENSIONS [*]
(extended abstract)

ANDREW CHI-CHIH YAO
Computer Science Department
Stanford University
Stanford, California 94305

ABSTRACT.

The problem of finding a minimum spanning tree connecting n points in a k-dimensional space is discussed under three common distance metrics -- Euclidean, rectilinear, and L_∞. It is shown that, for fixed $k \geq 3$, such a minimum spanning tree can be found in time $O(n^{2-a(k)}(\log n)^{1-a(k)})$, where $a(k) = 2^{-(k+1)}$. The bound can be improved to $O((n \log n)^{1.8})$ for points in 3-dimensional Euclidean space. The techniques used are applied to obtain $o(n^2)$ algorithms for finding a farthest pair in a set of n points and for other related problems.

1. INTRODUCTION.

Given an undirected graph with a weight assigned to each edge, a minimum spanning tree (MST) is a spanning tree whose edges have a minimum total weight among all spanning trees. Let E_p^k be the space of all k-tuples of real numbers with L_p -metric, i.e., the distance between two points x and y is given by $d_p(\tilde{x},\tilde{y}) = \left(\sum_i |x_i-y_i|^p\right)^{1/p}$, where x_i and y_i are components of $\tilde{x}$ and $\tilde{y}$. We shall agree that $d_\infty(\tilde{x},\tilde{y}) = \max_i |x_i-y_i|$. For any finite set of points V in E_p^k, let $G(V)$ denote the complete graph on V with a weight $d_p(\tilde{x},\tilde{y})$ assigned to the edge $\{\tilde{x},\tilde{y}\}$ for any $\tilde{x}, \tilde{y}$ in V.

An important application of the MST concept is in pattern recognition (see, e.g. [5]), where it is desired to find, for a given set V of n points in E_p^k, a MST of the associated graph $G(V)$. A simple algorithm is to compute $(d_p(\tilde{x},\tilde{y}))^p$ for all pairs $(\tilde{x},\tilde{y})$ in V, and use a MST algorithm for general graphs. This takes $O(n^2)$ total time, if k and p are fixed integers. However, as there are only kn input parameters, it is interesting to ask if there are better algorithms that run in $o(n^2)$ time in the worst case.

Several empirically good algorithms were studied by Bentley and Friedman [2], where a list of references to other applications of finding

---

[*] This research was supported in part by National Science Foundation grant MCS 72-03752 A03.

MST in k-dimensional spaces can be found. In the two-dimensional Euclidean case $(p = k = 2)$, Shamos and Hoey [6] gave a $O(n \log n)$ algorithm. No algorithm, however, is known to have a guaranteed worst-case running time of $o(n^2)$ when $k \geq 3$.

In this paper, asymptotically $o(kn^2)$ algorithms for MST are constructed. Fast algorithms for related geometric problems are also given using a similar technique.

2. A STUDY OF DOBKIN-LIPTON'S MULTIDIMENSIONAL SEARCH.

In [4] Dobkin and Lipton proved the following statement. Given a set of m hyperplanes $H_1, H_2, \ldots, H_m$ in E^k, it is possible to preprocess them in such a way that, when a new point $\tilde{x} \in E^k$ is subsequently given, it is possible to find out in $O(2^k \log m)$ time which region formed by the H_i's $\tilde{x}$ is in. The preprocessing procedure was described at a high level in [4] and the preprocessing cost has not been analyzed. We look into this problem with the following result.

Notation. $b(k) = 2^k$, $a(k) = 2^{-(k+1)}$.

Theorem 1. There is an algorithm which, for any m given hyperplanes $h_i(\tilde{x}) = 0$ $(1 \leq i \leq m)$, constructs in time $O(km^{b(k)})$ a data structure in such a way that, for any new input point $\tilde{x} \in E^k$, the region to which $\tilde{x}$ belongs can be found in $O(2^k \log m)$ time.

In the post office problem (called Nearest Neighbor problem in [1]), it is desired to preprocess n given points $\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_n$ in such a way that, when any new point is input, a nearest $\tilde{v}_i$ can be found quickly. It was noted in [4] that the post office problem, at least for Euclidean space, is a special case of the above multidimensional search problem. Since we have analyzed the general problem in Theorem 1, we can obtain the following result.

Theorem 2. There is an algorithm that preprocesses any n given points $\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_n$ of E_2^k in $O(k\, n^{2b(k)})$ time. For any new point $\tilde{x}$ given after the preprocessing, a nearest $\tilde{v}_i$ can be found in $O(2^k \log n)$ time. For the special case $k = 3$, the preprocessing time can be lowered to $O(n^6 \log n)$; alternatively it is possible to improve the preprocessing time to $O(n^5 \log n)$, but allow a search time of $O((\log n)^2)$ for this case $(k = 3)$.

For L_1 and L_∞ metrics, analogous results can be proved (but not the " $k = 3$ part" of Theorem 2).

An application of Theorem 1 follows:

Theorem 3. Let $c > 1$. Given m hyperplanes as in Theorem 1, we can preprocess the hyperplanes in time $O(m^c)$ in such a way that, for any new input $\tilde{x} \in E^k$, the region $\tilde{x}$ is in can be found in

$O\left(m^{1 - \frac{c-1}{b(k)-1}} (\log m)\right)$ time.

3. RESULTS FOR CONCRETE PROBLEMS.

Theorem 4. Let $k \geq 3$ be a fixed integer, $a(k) = 2^{-(k+1)}$, and all points to be considered are in E_p^k with $p \in \{1,2,\infty\}$. Then each of the following problems can be solved in time $O(n^{2-a(k)}(\log n)^{1-a(k)})$. For the case when $k = 3$ and $p = 2$, the bound can be improved to $O((n \log n)^{1.8})$.

MST-problem: Let V be a set of n points, find a minimum spanning tree on V.

NFN-problem (nearest foreign neighbor): Let $V_1, V_2, \ldots, V_\ell$ be disjoint sets of points, $V = \bigcup_i V_i$, and $|V| = n$. For each V_i and every $\tilde{x} \in V_i$, find a $\tilde{y} \in V - V_i$ such that $d_p(\tilde{x},\tilde{y}) = \min\{d_p(\tilde{x},\tilde{z}) \mid \tilde{z} \in V - V_i\}$.

GN-problem (geographic neighbor): Let V be a set of n points. For any $\tilde{x} \in V$, let $N(\tilde{x}) = \{\tilde{v} \mid v_i \geq x_i \text{ for all } 1 \leq i \leq k, \tilde{v} \neq \tilde{x}, \tilde{v} \in V\}$. For each $\tilde{x} \in V$, find a $\tilde{y} \in N(\tilde{x})$ such that $d_p(\tilde{x},\tilde{y}) = \min\{d_p(\tilde{x},\tilde{v}) \mid \tilde{v} \in N(\tilde{x})\}$ if $N(\tilde{x}) \neq \emptyset$.

AFP-problem [3] (all farthest points): Let V be a set of n points. For each $\tilde{x} \in V$, find a $\tilde{y} \in V$ such that $d_p(\tilde{x},\tilde{y}) = \max\{d_p(\tilde{x},\tilde{v}) \mid \tilde{v} \in V\}$.

FP-problem [3] (farthest pair): Let V be a set of n points, find $\tilde{x},\tilde{y} \in V$ such that $d_p(\tilde{x},\tilde{y}) = \max\{d_p(\tilde{u},\tilde{v}) \mid \tilde{u},\tilde{v} \in V\}$.

Theorem 5. The complexity of evaluating $(d_2(\tilde{x},\tilde{y}))^2$ for all pairs $(\tilde{x},\tilde{y})$ in a set V of n points in E_2^k is equivalent to the complexity of multiplying an $n \times k$ matrix and a $k \times n$ matrix. Similarly, the evaluation of all $d_\infty(\tilde{x},\tilde{y})$ is equivalent to matrix multiplication with $\{\min,+\}$ as operators instead of $\{+,*\}$.

Corollary. Suppose $k \leq n$. The minimum spanning tree for n points in E_2^k can be found in $O(n^2 k^{0.81})$ time, and for n points in E_∞^k can be found in $O\left(n^2 k \left(\frac{\log \log k}{\log k}\right)^{1/3}\right)$ time.

Theorem 5 can also be applied to yield similar bounds for all nearest neighbor or all farthest point problems, etc.

4. OPEN PROBLEMS.

(1) Improve the bounds obtained in the paper.

(2) Analyze the performance of new or existing fast heuristic algorithms for MST. For example, can one show that the AMST algorithm in [2] always constructs a spanning tree with length at most 5% over the true MST ?

(3) Prove bounds on average running time of MST algorithms for reasonable distributions.

REFERENCES

[1] J. L. Bentley, "Divide and conquer algorithms for closest point algorithms in multidimensional space," Ph.D. Thesis, Computer Science Department, University of North Carolina, December, 1976.

[2] J. L. Bentley and J. H. Friedman, "Fast algorithms for constructing minimal spanning trees in coordinate spaces," Stanford Computer Science Department Report STAN-CS-75-529, January, 1976.

[3] J. L. Bentley and M. I. Shamos, "Divide-and-conquer in multidimensional space," Proc. 8th Annual ACM Symp. on Theory of Computing (1976), 220-230.

[4] D. Dobkin and R. J. Lipton, "Multidimensional search problems," SIAM J. on Computing 5 (1976), 181-186.

[5] R. O. Dude and P. E. Hart, Pattern Classification and Scene Analysis, John Wiley, New York, 1973.

[6] M. I. Shamos and D. J. Hoey, "Closest-point problems," Proc. 16th Annual Symp. on Foundations of Computer Science (1975), 151-162.

FINDING SMALLEST SPANNING FORESTS AND TREES SUBJECT TO MANY DEGREE CONSTRAINTS

HAROLD N. GABOW
Department of Computer Science
University of Colorado at Boulder
Boulder, Colorado 80309

ABSTRACT

The degree-constrained spanning forest problem is to find a spanning forest with roots 1,...,n, where root i has degree d_i, and the cost is small as possible. This problem arises in communication network design; previous work is on the case n = 1. We give an algorithm that uses time $O(V^2E)$; if $n = O(1)$, the time is $O(\min(V^2, E \log\log V) + V \log V)$. The algorithm is based on an edge exchange technique. The degree-constraind spanning tree problem is to find a spanning tree where vertices i have degree d_i, for i = 1,...,n. This problem is NP-complete, even if $n = O(V^\varepsilon)$.

1. INTRODUCTION

We discuss two variants of the minimum spanning tree problem that arise in communication network design, the degree-constrained spanning forest (DCSF) and degree-constrained spanning tree (DCST) problems. These problems are related to the capacitated tree problem (CT) of network design. CT is intractable, in the sense of being NP-complete [P]. Here we show DCST is also intractable, even for a small number of degree-constraints; however, the closely related DCSF can be solved efficiently.

An instance of DCSF is specified by the following:

(i) G, an undirected graph. The vertices are numbered 1 to V; there are E edges;
(ii) c, a cost function assigning a real-valued cost to each edge;
(iii) n, the number of degree-constrained roots in the spanning forest. These roots are vertices 1,...,n.
(iv) d_i, the desired degree for root i, where $1 \le i \le n$.

If F is a spanning forest, let d(i) denote the degree of vertex i in F; let c(F) be the cost of F, i.e., $c(F) = \sum_{e \varepsilon F} c(F)$. Then the DCSF problem is to find a spanning forest F with roots i, for i = 1,...,n, where each degree $d(i) = d_i$; further, c(F) is as small as possible, subject to these constraints.

In a communication network, a root represents a computing site, and a non-root vertex is a terminal. Each terminal must be linked to some (arbitrary) computing site by a cable path; thus we seek a spanning forest. Each computing site has a number of communication ports; thus each root has a degree constraint. To use as little cable as possible, we seek a smallest cost forest. This gives DCSF.

Now suppose the computing sites must communicate with each other. If this portion of the network is designed in a separate stage, DCSF still models the design problem for the terminals. However if a single connected network is to be designed at once, we seek a spanning tree, where some vertices have degree constraints, and the cost is small as possible. This is the DCST problem.

Previous work on DCSF and DCST concentrates on the case n = 1, i.e., one vertex has a degree constraint (in this case, DCSF and DCST are identical). Glover and Klingman [GK] present an algorithm that is efficient when the desired degree is small; in general it has an $O(V^2)$ time bound (the time estimate is ours). A similar algorithm [G] is more efficient when the desired degree is large; it has an $O(\min(V^2, E \log \log V) + V \log V)$ time bound. We generalize these algorithms to the DCSF problem with $n > 1$. If $n = O(1)$, we achieve the above time bound $O(\min(V^2, E \log \log V) + V \log V)$; in general, the time is $O(V^2 E)$. We also show DCST is NP-complete even if $n = O(V^\varepsilon)$, for any $\varepsilon > 0$.

We make some more preliminary remarks. The above notation defining DCSF is used throughout the paper. Also, let N be the set of roots, i.e., $N = \{1, \ldots, n\}$. A forest F is rooted at N if its roots are exactly N. We assume G has no edges (i,j), joining two roots i,j ($1 \le i,j \le n$), for such edges are not in a spanning forest rooted at N. The components of F are the trees obtained by removing all edges incident to N from F. Hence a component C is a connected component of a set $F - (i,j)$, where $i \le n$, $j > n$, $j \in C$.

The rest of the paper is organized as follows. Section 2 gives graph theoretic results for DCSF. Section 3 presents two algorithms for DCSF. Finally Section 4 gives the NP-completeness result for DCST.

2. EXCHANGE SEQUENCES

This section defines "increasing" and "decreasing exchange sequences", and derives their basic properties.

Let $\delta = (\delta_1, \ldots, \delta_m)$ be a vector of m non-negative integers δ_i, where $m \le n$. Then F_δ denotes the set of all spanning forests rooted at N, such that $d(i) = \delta_i$, for all i in $1 \le i \le m$. Let δ^+ be the vector $(\delta_1, \ldots, \delta_{m-1}, \delta_m + 1)$, and δ^- be $(\delta_1, \ldots, \delta_{m-1}, \delta_m - 1)$. We will study the relation between smallest forests in F_δ and F_{δ}, F_{δ^-}.

An edge exchange for a forest F rooted at N is an ordered pair of edges e, e', where $e \in F$, $e' \notin F$, and $F - e + e'$ is a forest rooted at N. The last condition can be restated in two equivalent ways: (i) In the forest $F - e$, edge e' joins the unique tree not containing a vertex of N to another tree (containing a vertex of N). (ii) In $F + e'$, edge e is on one of the two paths joining an end of e' to a root in N.

If e is incident to vertex i and e' is not, then exchange e, e' decreases the degree $d(i)$ by 1. In general, to convert a forest in F_δ to one in F_{δ^-}, we use a sequence of such exchanges. A decreasing exchange sequence (des), illustrated in Fig. 1(a), is a sequence of edge pairs e_i, e'_i, for $i = 1, \ldots, p$, where $p \le m$, $F - \{e_i\}_1^p + \{e'_i\}_1^p$ is a forest rooted at N, and the edges have the form

$$e_i = (r_i, s_i),\ e'_i = (r_{i+1}, s'_i) .$$

Here $r_1 = m$, and for $1 \le i \le m$, $r_i \le m$; r_{m+1}, s_i, $s'_i > m$, and all of these vertices are distinct. A des is "decreasing" in the sense that if $F \in F_\delta$, then the new forest is in F_{δ^-}.

An increasing exchange sequence (ies) is defined similarly, except we require $F - \{e'_i\}_1^p + \{e_i\}_1^p$ to be a forest. This is illustrated in Fig. 1(b). The sequence is "increasing" since the new forest is in F_{δ^+}.

We construct exchange sequences from exchanges for F, as follows.

Lemma 1: For $1 \le i \le p$, let e_i, e'_i be an edge exchange for F, where all edges are distinct. Let $e_i = (r_i, s_i)$, $e'_i = (r'_i, s'_i)$, where $s_i, s'_i > n$. For $1 \le i < p$, let $r_i, r'_i \le m$; for $i = p$ assume one of the following:

(a) $r_p, r'_p \le m$;

(b) $r_p \le m, r'_p > m$;

(c) $r_p > m, r'_p \le m$, and further, if s'_p is in the same component (of F) as some vertex s_i, $1 \le i < p$, then the path (in F) from s'_p to s_p is disjoint from the path from s'_i to s_i (except for possibly s_p).

Then $F - \{e_i\}_1^p + \{e'_i\}_1^p$ is a forest rooted at N.

Proof: If both $r_i, r'_i \varepsilon N$, the exchange e_i, e'_i just transfers the component of s_i from the tree rooted at r_i to the tree rooted at r'_i. Thus the Lemma holds if $r_i, r'_i \le n$. In particular, case (a) is true. Case (b) follows similarly, since the p^{th} exchange e_p, e'_p transfers the component of s_p to whatever tree contains vertex s'_p .

Now we discuss case (c). By the above remark we may assume $r_p \notin N$. Let the component of s'_p (in F) be joined to its root vertex r by the edge (r,s). Sine e_p, e'_p is an edge exchange (for F), the path in F from s'_p to r contains edge e_p .

Suppose (r,s) is not among the edges e_i, $1 \le i < p$. Then e_p, e'_p is still a valid exchange after the first $p-1$ exchanges have been made. So the Lemma holds for this possibility.

Next suppose $(r,s) = (r_i, s_i)$, for some i in $1 \le i < p$. This case is illustrated in Fig. 2. After the i^{th} exchange, the component containing s'_p is joined by edge e'_i to its new root, r'_i . The hypothesis (c) implies that the path from s'_p to root r'_i starts with the path (in F) from s'_p to s_p, and ends with a portion of the path from s_i to r'_i. Edge e_p is in this path. Hence e_p, e'_p is still a valid exchange. Thus the Lemma holds for this possibility. □

Now we show smallest forests in F_δ and $F_{\delta+}, F_{\delta-}$ are related by exchange sequences.

Theorem 1: Let F be a smallest forest in F_δ .

(a) Assume $F_{\delta-} = \emptyset$. Then there is a des e_i, e'_i, $i = 1,\ldots,p$, such that $F - \{e_i\}_i^p + \{e'_i\}_1^p$ is a smallest forest in $F_{\delta-}$.

(b) Assume $F_{\delta+} \ne \emptyset$. Then there is an ies e_i, e'_i, $i = 1,\ldots,p$, such that $F - \{e_i\}_1^p + \{e'_i\}_1^p$ is a smallest forest in $F_{\delta+}$.

Proof: We prove part (a). The argument for (b) is analogous.

Let F' be a smallest forest in $F_{\delta-}$ containing as many edges of F as possible. We define e_i, e'_i so e_i, e'_i is a des for F, and e'_i, e_i is an *ies* for F'. We do this inductively, as follows.

Suppose, for some $j \geq 1$, e_i, e'_i is defined for $1 \leq i < j$, such that e_i, e'_i is an exchange for F; e'_i, e_i is an exchange for F'; $e_i = (r_i, s_i)$, $e'_i = (r_{i+1}, s'_i)$, where $r_1 = m$, and for $1 \leq i < j$, r_i, $r_j \leq m$, $s_i, s'_i > n$, and all vertices are distinct. Now we define e_j, e'_j.

First we define edge $e_j = (r_j, s_j)$. If $j = 1$, vertex $r_j = m$; if $j > 1$, vertex r_j is known from edge $e'_{j-1} = (r_j, s'_{j-1})$. Choose vertex s_j so edge $e_j \in F - F'$. Note such an e_j exists. For if $j = 1$, the degree of r_j is greater in F than in F'; if $j > 1$, the degree of r_j is the same in F and F', and since $e'_{j-1} = (r_j, s'_{j-1}) \in F' - F$, an edge $e_j \in F - F'$ exists.

Now we define edge $e'_j = (r_{j+1}, s'_j)$. This is illustrated in Fig. 3. Let r be the root of the tree in F' containing s_j. Consider the path in F' from s_j to r. Let e'_j be the first edge in this path that joins the component (of F) containing s_j to some tree of F. Such an edge exists, since the path ends outside of s_j's component.

These definitions imply e_j, e'_j is an exchange for F, and e'_j, e_j is an exchange for F'. It is easy to check vertices s_j and s'_j are distinct from all others. Now we consider three cases.

Case I: $r_{j+1} = r_k$, for some k in $1 \leq k \leq j$. We show this case is impossible, by contradicting the definition of F'. Lemma 1(a) implies $F - \{e_i\}^j_k + \{e'_i\}^j_k$ is a forest rooted at N. Then Case I implies this forest is in $\mathcal{F}_\delta$. Similarly, Lemma 1(a) implies $F' - \{e'_i\}^j_k + \{e_i\}^j_k$ is in $\mathcal{F}_{\delta-}$. Since F is a smallest forest in $\mathcal{F}_\delta$, $c(F) - \sum_k^j c(e_i) + \sum_k^j c(e'_i) \geq c(F)$; similarly $c(F') - \sum_k^j c(e'_i) + \sum_k^j c(e_i) \geq c(F')$. These inequalities imply equality holds in both cases. Thus $F' - \{e'_i\}^j_k + \{e_i\}^j_k$ is a smallest cost forest in $\mathcal{F}_{\delta-}$, containing more edges of F than F' does. This is the desired contradiction.

Case II: $r_{j+1} \leq m$, $r_{j+1} \neq r_k$, for k in $1 \leq k \leq j$. Here we make $p > j$, and repeat the above procedure to define the exchange e_{j+1}, e'_{j+1}.

Case III: $s_{j+1} > m$. Let $p = j$. Then Lemma 1(b) implies $F - \{e'_i\}^p_1 + \{e'_i\}^p_1$ is a forest rooted at N. So this forest is in $\mathcal{F}_{\delta-}$, and e_i, e'_i is a *des*. Since F' is a smallest forest in $\mathcal{F}_{\delta-}$, we have

$$c(F) - \sum_1^p c(e_i) + \sum_1^p c(e'_i) \geq c(F') . \tag{1}$$

Now we check $F' - \{e'_i\}^p_1 + \{e_i\}^p_1$ is a forest rooted at N, by verifying condition (c) of Lemma 1. Consider the last exchange $e'_p = (r_{p+1}, s'_p)$,

$e_p=(r_p,s_p)$. Suppose s_p is in the same tree of F' as some vertex s_i', $1 \le i < p$. By construction, the path in F' from s_i to s_i' is contained in the component of F containing s_i; the path in F' from s_p to s_p' is contained in the component of F containing s_p. These components are distinct, by the distinctness of vertices s_i, s_p. Thus condition (c) is satisfied.

Hence $F' - \{e_i'\}_1^p + \{e_i\}_1^p \in \mathcal{F}_\delta$, and since F is a smallest forest in $\mathcal{F}_\delta$, $c(F') - \sum_1^m c(e_i') + \sum_1^m c(e_i) \ge c(F)$. This inequality, plus (1), imply equality holds in both cases. Hence $F - \{e_i\}_1^p + \{e_i'\}_1^p$ is a smallest forest in $\mathcal{F}_{\delta-}$, and e_i, e_i' is the desired *des*. □

Now we draw some useful conclusions from Theorem 1. First we consider when $\mathcal{F}_\delta$ is non-empty.

Corollary 1: Suppose $\mathcal{F}_\delta \ne \emptyset$. Let $\delta' = (\delta_1', \ldots, \delta_m')$, and suppose $\mathcal{F}_{\delta'} \ne \emptyset$.

(a) If $\delta_i \ge \delta_i'$, for $1 \le i < m$, and $\delta_m > \delta_m'$, then $\mathcal{F}_{\delta-} \ne \emptyset$.

(b) If $\delta_i \le \delta_i'$, for $i \le i < m$, and $\delta_m < \delta_m'$, then $\mathcal{F}_{\delta+} \ne \emptyset$.

Proof: Similar to Theorem 1. □

The next result restates Theorem 1 in a more useful way. Define the *cost* of the exchange sequence e_i, e_i', $i = 1, \ldots, p$, as $\sum_{i=1}^{p} c(e_i') - c(e_i)$. Note the cost of the exchange is the difference between the cost of the new forest and the old. A *min cost des* or *ies* has smallest cost possible.

Corollary 2: Let F be a smallest forest in $\mathcal{F}_\delta$.

(a) Assume $\mathcal{F}_{\delta-} \ne \emptyset$. Let e_i, e_i', $i = 1, \ldots, p$ be a min cost *des*. Then $F - \{e_i\}_1^p + \{e_i'\}_1^p$ is a smallest forest in $\mathcal{F}_{\delta-}$.

(b) An analogous result holds for $\mathcal{F}_{\delta+}$. □

Now we investigate the situation where the degrees of vertices i, $m < i \le n$, must be as small or large as possible. For an m-vector δ, define *$s(\delta)$* as the lexicographically smallest, $n-m$ vector, such that $\mathcal{F}_{\delta,s(\delta)} \ne \emptyset$. (Note "$\delta,s(\delta)$" denotes the concatenation of vectors δ, $\ell(\delta)$ into an n-vector). Similarly, define $\ell(\delta)$ to be the lexicographically largest n-m vector such that $\mathcal{F}_{\delta,\ell(\delta)} \ne \emptyset$.

Lemma 2: (a) Let F be a smallest cost forest in $\mathcal{F}_{\delta,\ell(\delta)}$. Assume $\mathcal{F}_{\delta-} \ne \emptyset$. Then there is a *des* e_i, e_i', $i = 1, \ldots, p$, such that $F - \{e_i\}_1^p + \{e_i'\}_1^p$ is a smallest cost forest in $\mathcal{F}_{\delta-,\ell(\delta-)}$. If $\mathcal{F}_{\delta+} \ne \emptyset$, there is an *ies* e_i, e_i', $i = 1, \ldots, p$, such that $F - \{e_i\}_1^p + \{e_i'\}_1^p$ is a smallest cost forest in $\mathcal{F}_{\delta+,\ell(\delta+)}$.

(b) An analogous result holds for $\mathcal{F}_{\delta,s(\delta)}$.

Proof: Use Theorem 1, with edge costs appropriately modified. □

To find smallest forests in $\mathcal{F}_{\delta,\ell(\delta)}$, define a *lex max min cost des* as a *des* e_i, e'_i, $i = 1,\ldots,p$, of smallest possible cost, subject to the restriction that the last edge e'_p is incident to the lowest possible numbered root j, $m < j \le n$. (Note if no *des* ends at a root j, there is no restriction). A *lex max min cost ies*, and *lex min min cost des* and *ies*, are all defined similarly.

Corollary 3: (a) Let F be a smallest forest in $\mathcal{F}_{\delta,\ell(\delta)}$. Assume $\mathcal{F}_{\delta^-} \ne \emptyset$. Let e_i, e'_i, $i = 1,\ldots,p$, be a lex max min cost *des*. Then $F - \{e_i\}_1^p + \{e'_i\}_1^p$ is a smallest forest in $\mathcal{F}_{\delta^-,\ell(\delta^-)}$. If $\mathcal{F}_{\delta^+} \ne \emptyset$, let e_i, e'_i, $i = 1,\ldots,p$, be a lex max min cost *ies*. Then $F - \{e_i\}_1^p + \{e'_i\}_1^p$ is a smallest forest in $\mathcal{F}_{\delta^+,\ell(\delta^+)}$.

(b) An analogous result holds for $\mathcal{F}_{\delta,s(\delta)}$. □

3. FOREST ALGORITHMS

This section presents two algorithms for finding optimal degree-constrained forests.

The basic step in each algorithm is finding a lex min or max, min cost, *ies* or *des*. We treat this as a shortest path problem, as follows. Consider the case of an *ies*. We define a directed graph G_I. The vertices of G_I are $1,2,\ldots,n,n+1$. We think of the vertex v in G as corresponding to the vertex $\min(v,n+1)$ in G_I. (So a root corresponds to itself, and all other vertices in G correspond to $n+1$.) An edge in G_I represents an edge exchange: G_I contains edge (i,j) exactly when G contains vertices i,v,w,i', such that v and w are in the same component of F, and edges $(i,v) \notin F$, $(i',w) \in F$, $i \ne i'$, and $j = \min(i',n+1)$. (Thus $(i',w),(i,v)$ is an edge exchange.) Edge lengths are defined by setting

$$\ell(i,j) = \min\{c(i,v) - c(i',w) \mid (i,j) \text{ represents exchange } (i',w),(i,v)\}.$$

Thus $\ell(i,j)$ is the smallest cost of an exchange that increases $d(i)$ by 1 and, if $j \le n$, decreases $d(j)$ by 1.

It is easy to see an *ies* for $(\delta_1,\ldots,\delta_m)$ corresponds to a path in G_I starting at vertex m and ending at some j, $m < j \le n+1$. The cost of the exchange sequence is the length of the path. The exchange sequence is lex max if j is as large as possible, and lex min if j is as small as possible; it is min cost if the path has shortest length possible. Thus finding an *ies* is a shortest path problem in G_I.

The graph G_D for *des*'s is defined analogously.

Now we show how to construct G_I.

Algorithm BI

```
for each vertex $v > n$, let $R(v)$ be the edge joining $v$'s com-
ponent to a root, and let $C(v)$ be the edge with largest cost on
the path in $v$'s component from $v$ to $R(v)$. ($C(v)$ is unde-
fined if this path is empty);
for i: = 1 to n do
    begin for each edge $(i,v)$ do
        begin let $R(v) = (i',w)$, $i' \le n$ ;
```

```
5.          if $i \neq i'$ then update $G_I$ so it contains edge $(i,i')$ with
              length $c(i,v) - c(i',w)$ or less;
6.          if $C(v)$ is defined then update $G_I$ so it contains edge
              $(i,n+1)$ with length $c(i,v) - c(C(v))$ or less;
      end end;
```

In our time and space estimates, V, E, and E_N represent the number of vertices, edges, and edges incident to N (all in graph G), respectively.

Lemma 3: Algorithm BI builds the graph G_I in time $O(V+E_N)$ and space $O(V+E)$.

Proof: For the time bound, note line 1 is $O(V)$, using a depth-first traversal of F. The remaining details are left to the reader. □

The algorithm to build G_D is analogous to BI.

Now we present an algorithm for optimum degree-constrained forests, based on ies's.

Algorithm I

```
1.  let $S$ be a smallest spanning forest in $G-N$;
2.  let $F$ be a smallest forest in $\mathcal{F}_\alpha$, where $\alpha$ is the lexico-
      graphically smallest n-vector such that $\mathcal{F}_\alpha \neq \emptyset$;
3.  for $m := 1$ to $n$ do
4.      begin while $d(m) < d_m$ do
5.          begin form the graph $G_I$, with edge lengths $\ell$;
6.              find a path from $m$ to the lowest possible vertex $j$,
                    $m < j \le n+1$, with shortest length possible;
7.              let the corresponding ies be $e_i, e_i'$, $i = 1,\ldots,p$;
8.              $F \leftarrow F - \{e_i\}_1^p + \{e_i'\}_1^p$;
          end end;
```

Theorem 2: Algorithm I finds a smallest cost forest rooted at N with $d(i) = d_i$, $1 \le i \le n$. The time is $O(\min(V^2, E \log\log V) + (\sum_{i=1}^{n} d_i)(V + E_N + \min(n^3, n E_N))$ and the space is $O(V+E)$.

Proof: For the time bound, note line 1 requires $O(\min(V^2, E \log\log V))$ [CT,Y]. Line 6 is the single source, shortest path problem in a graph with negative edges. In general, the time is $O(VE)$ [L]. Since G_I has $O(\min(n^2, E_N))$ edges, line 6 is $O(\min(n^3, n E_N))$. □

Corollary 4: Algorithm I uses time $O(V^2E)$. □

Algorithm I is particularly efficient when the degree constraints are small. For example, if $\sum_{i=1}^{n} d_i = O(1)$ (whence $n = O(1)$), the time is just the time to find the initial spanning forest, $O(\min(V^2, E \log\log V))$.

Now we discuss an algorithm based on des's. It differs from the previous one in a number of ways. It begins by pruning the graph G to a small set of edges, using this fact:

Corollary 5: Let S be a smallest cost spanning forest in graph G-N. Let E_N be the set of edges of G incident to N. Then $S \cup E_N$ contains a smallest cost forest in any non-empty set $\mathcal{F}_\delta$.

Proof: Observe the edges that Algorithm I adds to F. □

After each exchange sequence is done, the algorithm updates the auxiliary graph G_D. For this, a system of priority queues representing the edges in $S \cup E_N$ is used. The system consists of one family of lists L, and three families of priority queues, Q. In the definitions below, C is a component of F, and i and i' are roots, $1 \le i, i' \le n$.

$L(C) = \{e \mid e$ is the smallest edge joining C to some root $i'\}$;
$Q(C) = \{e \mid e$ is the smallest edge joining C to some other component $C'\}$;
$Q(i,i') = \{e \mid e$ is the smallest edge joining some component C, with root i, to $i'\}$;
$Q(i,n+1) = \{e \mid e$ is the smallest edge joining some component C, with root i, to another component$\}$;

The smallest edge in each queue $Q(i,i')$, $Q(i,n+1)$ corresponds to an edge of G_D.

Lemma 4: G_D can be updated after the exchange sequence e_i, e_i', $i = 1,\ldots,p$, in time $O(\min(n^2,E_N)\log V)$ and space $O(V+E)$. ☐

Note the analogous system of queues to update the graph G_I is not as efficient. The difficulty occurs in the last exchange, e_p, e_p'. In a des, this exchange merges two components; in an ies, it splits one component. Priority queues handle the merge efficiently, but not the split. So the update approach is not efficient for ies's.

Now we present the algorithm based on des's.

Algorithm D

```
 1.  let S be a smallest spanning forest for G-N;
     let $E_N$ be the set of all edges incident to vertices in N;
 2.  remove all edges from G except $S \cup E_N$;
 3.  let F be a smallest spanning forest in $\mathcal{F}_\beta$, where $\beta$ is the
     lexicographically largest n-vector such that $\mathcal{F}_\beta \ne \emptyset$;
 4.  let B be a list of all roots i, $i \le i \le n$, with $d(i) > d_i$;
 5.  form the graph $G_D$, with edge lengths;
 6.  while B is non-empty do
 7.     begin remove a vertex m from B;
 8.        while $d(m) > d_m$ do
 9.           begin find a path from m to the lowest possible vertex
                 j in $\{i \mid 1 \le i \le n$ and $d(i) < d_i\} \cup \{n+1\}$, with
                 shortest length possible;
10.              let the corresponding des be $e_i, e_i'$, $i = 1,\ldots,p$;
11.              $F \leftarrow F - \{e_i\}_1^p + \{e_i'\}_1^p$;
12.              update the graph $G_D$;
     end end;
```

Theorem 3: Algorithm D finds a smallest cost forest rooted at N with $d(i) = d_i$, $1 \le i \le n$. The time is $O(\min(V^2, E \log\log V) + (V - \sum_{i=1}^{n} d_i)\min(n^2, E_N)(n + \log V))$ and the space is $O(V+E)$.

Proof: The main step in the correctness proof is an induction showing a forest F constructed by the algorithm is a smallest forest in $\mathcal{F}_{\delta,\ell(\delta)}$. Here δ is defined as follows: Number the roots of G_D from 1 to n, so roots i with $d(i) \ge d_i$ are numbered $1,\ldots,m$, for some m; roots i with $d(i) < d_i$ are numbered $m+1,\ldots,n$, and the original relative order of these roots is preserved. □

Algorithm D achieves the time bound of Corollary 4, $O(V^2E)$, in the general case. We also note two special cases. If $n = O(1)$, the time is $O(\min(V^2, E \log\log V) + V \log V)$. If the degree constraints are very large, i.e., $V - \sum_{i=1}^{n} d_i = O(1)$, the time is $O(VE)$.

4. DCST IS NP-COMPLETE

This section gives a simple proof that finding a spanning tree subject to even a small number of degree constraints is NP-complete.

In our definition of DCST, we ignore edge costs, since just meeting the degree constraints is a difficult problem. So the DCST problem is: For an undirected graph G, with vertices numbered 1 to V, is there a spanning tree such that $d(i) = d_i$, for $1 \le i \le n$?

The proof makes use of the Hamiltonian path problem (HP): For an undirected graph G, with vertices numbered 1 to V, is there a path from 1 to V including all other vertices exactly one? HP is NP-complete [AHU].

Theorem 4: For any positive ε, DCST with $n = O(V^{\varepsilon})$ is NP-complete.

Proof: We give a polynomial transformation of HP into the desired version of DCST. Let the graph G, with vertices 1 to V, be an instance HP. Let $\bar{V} = V^{\lceil 1/\varepsilon \rceil}$. Construct a graph $\bar{G}$ with $\bar{V} = V^{\lceil 1/\varepsilon \rceil}$ vertices, by adding to G new vertices $x = V+1, \ldots, \bar{V}$, and new edges (x, V). Consider the instance of DCST on graph $\bar{G}$ with constraints $d_1 = 1$, $d_i = 2$ for i in $1 < i < V$. It is easy to see this problem satisfies the restriction on n, and has the same yes-no answer as the original instance of HP. □

ACKNOWLEDGMENT

The author thanks Ron Rivest for his communication on NP-completeness results.

REFERENCES

[AHU] A. Aho, J. Hopcroft and J. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, Mass., 1974.

[CT] D. Cheriton and R. E. Tarjan, "Finding minimum spanning trees," SIAM J. on Computing 5 (1976), pp. 724-741.

[G] H. N. Gabow, "A good algorithm for smallest spanning trees with a degree constraint," Report CU-CS-105-77, Dept. of Computer Science, University of Colorado, Boulder, CO, 1977; also Networks, to appear.

[GK] F. Glover and D. Klingman, "Finding minimum spanning trees with a fixed number of links at a node," Report No.74-5, Research Report CS #169, Center for Cybernetic Studies, University of Texas at Austin, Texas, 1974.

[L] E. L. Lawler, Combinatorial Optimization: Networks and Matroids, Holt, Rinehart, and Winston, New York, 1976.

[P] C. H. Papadimitriou, "The complexity of the capacitated tree problem," Tech. Rep. TR-21-76, Center for Research in Comp. Technology, Harvard U., Cambridge, Mass., 1976.

[Y] A. C. Yao, "An O(|E| log log |V|) algorithm for finding minimum spanning trees," Information Processing Letters 4 (1975), pp. 21-23.

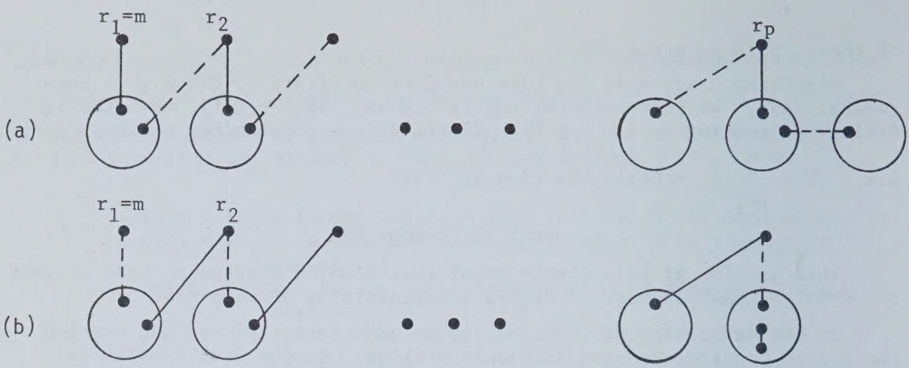

$r_1=m$
r_2
r_p
(a)
$r_1=m$
r_2
(b)

Figure 1 (a) Des (b) Ies.

Key: ●————● F edge

●– – –● edge not in F.

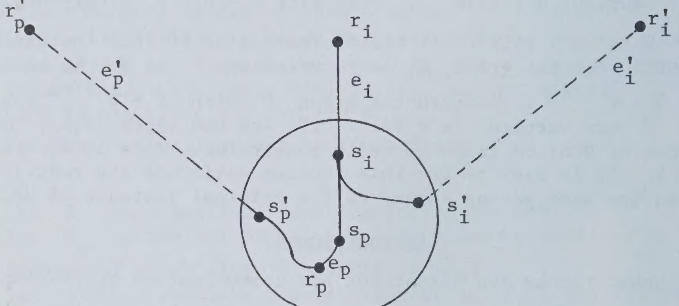

r'_p
e'_p
r_i
e_i
r'_i
e'_i
s_i
s'_p
s'_i
s_p
e_p
r_p

Figure 2: Case (c) of Lemma 1.

———— F edge.

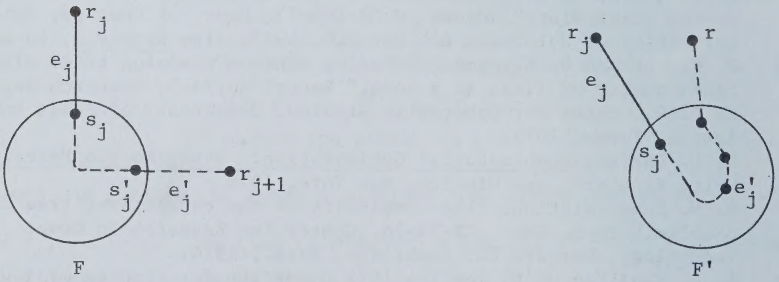

r_j
e_j
s_j
s'_j
e'_j
r_{j+1}
F
r_j
e_j
r
s_j
e'_j
F'

Figure 3: Definition of edges e_j, e'_j in Theorem 1.

———— F edge

------- F' edge

COMPUTATIONAL COMPLEXITY OF k-MAXIMAL CUTS

TAKAO NISHIZEKI*, HIDENAO OGAWA and NOBUJI SAITO
Faculty of Engineering, Tohoku University
Sendai, Japan 980

Abstract: A concept "k-maximal" is introduced as a measure to show the structural stability of approximate solutions of maximization problems. Characterizations of and algorithms for k-maximal cuts of graphs are given, and the computational complexity is investigated.

1. INTRODUCTION

Maximum cut problems, in which one would like to find a cut of a maximum cardinality of a given graph, arise in many applications, including printed circuit design [1]. Recent work has shown that this problem belongs to the class of NP-complete problems [2-4], and therefore it seems unlikely that any polynomial time bounded algorithm exists. Takanami has proposed an algorithm with elementary tree transformations [5]. Sahni and Gonzalez [6] and Kubo et. al. [7] have given approximation algorithms for the problem.

In this paper we introduce a new concept "k-maximal" as a measure to show the structural stability of approximate solutions of maximization problems, and give the characterizations of k-maximal cuts as an example of its applications. Furthermore we give algorithms to find a k-maximal cut of a graph $G=(V,E)$ in time $O(|V|^k.|E|^2)$ and especially a 1-maximal cut in time $O(|E|^2)$. The effectiveness of the algorithms is verified by computer simulation on random graphs.

2. PRELIMINARY DEFINITIONS

First we define "k-maximal" as a generalized version of "maximal" as follows:

Definition 1. Let $\mathcal{A}$ denote the family of subsets of a set A which satisfy a certain property. A subset $A_1 \in \mathcal{A}$ is k-maximal with respect to the property if there exists no subset

---

*Visiting at Dept. of Math, Carnegie-Mellon Univ., Pittsburgh, PA 15213. This research was supported in part by Murata Oversea Schalorship Foundation and the Grant in Aid for Scientific Research of the Ministry of Education, Science and Culture of Japan under Grant: Cooperative Research (A) 135017 (1976).

$A_2 \in \mathcal{A}$ such that $|A_2| > |A_1|$ and $|A_1 - A_2| \leq k$. Here k is an integer.

Note that "0-maximal" is equivalent to "maximal" in common use, and every subset in $\mathcal{A}$ is k-maximal if k is negative.

In this paper we deal with finite undirected connected graphs. $G=(V,E)$ denotes a graph with the vertex set V and the edge set E. A cutset of a graph $G=(V,E)$ is a minimal subset of E whose removal results in a disconnected or trivial graph.

Definition 2. A cut of a graph is a cutset or a mod 2 sum of some cutsets. In other words a cut K of a graph $G=(V,E)$ is a subset of E such that $K = \{(u,v) \mid u \in V_1,\ v \in \overline{V}_1$ for a nonempty subset $V_1 \subsetneq V\}$. Here a mod 2 sum of sets A_1 and A_2, denoted by $A_1 \oplus A_2$, means $(A_1 \cup A_2) - (A_1 \cap A_2)$, and $\overline{A}_1$ the complement of A_1.

Note that a cut K of a graph G can be represented as follows [8].

$$K = \sum_{i=1}^{n} S_i , \qquad (1)$$

where $S_1, S_2, \ldots, S_n$ are cutsets of G such that $S_i \cap S_j = \emptyset$ for $i \neq j$. Here we use the following notation: $A_1 \cup A_2 \cup \ldots \cup A_n = A_1 + A_2 + \ldots + A_n = \sum_{i=1}^{n} A_i$ if $A_i \cap A_j = \emptyset$.

Let $S(e,T)$ denote the fundamental cutset determined by a tree T and an edge $e \in T$, i.e., the unique cutset contained in $\overline{T} \cup \{e\}$. Note that any cut K is represented as follows [8]:

$$K = \bigoplus_{e \in K \cap T} S(e,T) . \qquad (2)$$

3. CHARACTERIZATIONS OF k-MAXIMAL CUTS

We present the following five theorems characterizing k-maximal cuts.

Theorem 1. The following statements are equivalent for a graph $G=(V,E)$ and a cut K of G.

(a) A cut K is 0-maximal (i.e., maximal).

(b) There exists no cutset S such that $K \cap S = \emptyset$.

(c) There exists a (spanning) tree T such that $T \subset K$. (Here T is the edge set of the tree.)

(d) The graph $G - \overline{K}$ (which is obtained from G by deleting the edges in $\overline{K}$) is connected.

The part, (a) ⇔ (d), of Theorem 1 has appeared in [7].

Theorem 2. A subset K of E is a 0-maximal cut of a graph $G=(V,E)$ if and only if there exists a tree T such that $K = \bigoplus_{e \in T} S(e,T)$.

Theorem 3. Let K be a (k-1)-maximal cut of a graph $G=(V, E)$, where $k \geq 1$.

(a) K is k-maximal if and only if there exists no cutset S such that $k = |K \cap S| < |S - K|$, and

(b) if there exists a cutset S such that $k = |K \cap S| < |S - K|$, then $K \cap S$ is a cutset of the graph $G_K = G - \overline{K}$.

(Proof) (a) Necessity: If there exists a cutset S such that $k = |K \cap S| < |S - K|$, then $K_1 = K \oplus S$ is a cut by Definition 2. Since $|K_1| > |K|$ and $|K - K_1| = k$, K is not k-maximal by Definition 1.

Sufficiency: If K is not k-maximal, then there exists a cut K_1 such that $|K_1| > |K|$ and $|K - K_1| = k$ by Definition 1. Since $K_2 = K \oplus K_1$ is a cut, by (1) K_2 can be represented as follows: $K_2 = \sum_{S_i \in \mathcal{S}' \subset \mathcal{S}} S_i$, where $\mathcal{S}$ is the set of all the cutsets of G. It is clear that

$$k = |K - K_1| = |K \cap K_2| < |K_2 - K| = |K_1 \cap K_2| .$$

Therefore we shall show that K_2 is a cutset. Suppose $|\mathcal{S}'| \geq 2$. Then there exists a cutset $S_i \in \mathcal{S}'$ such that

$$k > |K \cap S_i| < |K_1 \cap S_i| = |S_i - K| .$$

Since the cut $K_3 = K \oplus S_i$ satisfies $|K_3| > |K|$ and $|K - K_3| = |K \cap S_i| < k$, K is not (k-1)-maximal, contrary to the assumption. Therefore $|\mathcal{S}'| = 1$, that is, K_2 is a cutset.

(b) If there exists a cutset S such that

$$k = |K \cap S| < |S - K|, \tag{3}$$

then $K \cap S$ is a cut of the graph G_K [9]. By (1) $K \cap S$ can be represented as follows:

$$K \cap S = \sum_{S_{Ki} \in \mathcal{S}'_K \subset \mathcal{S}_K} S_{Ki} ,$$

where $\mathcal{S}_K$ denotes the set of all the cutsets of the graph G_K. We shall show that $|\mathcal{S}'_K| = 1$. Suppose that $|\mathcal{S}'_K| \geq 2$, then there exists a cutset $S_{Ki} \in \mathcal{S}'_K$ such that $1 \leq |S_{Ki}| < k$. There must exist a cutset S_1 of G such that

$$S_{Ki} = K \cap S_1 . \tag{4}$$

Since K is (k-1)-maximal, by (a)

$$|S_{Ki}| = |K \cap S_1| \geq |S_1 - K| . \tag{5}$$

A cut $K_1 = S \oplus S_1$ satisfies the following:

$$1 \leq |K \cap K_1| = |K \cap S| - |S_{Ki}| < k . \tag{6}$$

Furthermore

$$K_1 - K = ((S - K) - (S_1 - K)) + (S_1 - S) . \tag{7}$$

By the definition of a cutset

$$S_1 - S \neq \emptyset . \tag{8}$$

By (3)-(8)

$$\begin{aligned} |K_1 - K| &> |S - K| - |S_1 - K| \\ &> |K \cap S| - |S_{Ki}| = |K \cap K_1| . \end{aligned} \tag{9}$$

$K_2 = K \oplus K_1$ is a cut, and by (6) and (9) $|K_2| > |K|$ and $|K - K_2| = |K \cap K_1| < k$. Therefore K is not (k-1)-maximal, contrary to the assumption. Hence $|\mathcal{S}'_K| = 1$, that is, $K \cap S$ is a cutset of G_K. Q.E.D.

Theorem 4. Let $K \subset E$ be a (k-1)-maximal cut of a graph $G = (V,E)$, and let $k \geq 1$. Then K is k-maximal if and only if in the graph $G_K = G - \overline{K}$ there exists no cutset $S_1 = \{(u,v) \mid u \in V_1, v \in \overline{V}_1\}$ of cardinality k satisfying the following augmenting condition: There exist at least k + 1 edges $(u_i, v_i) \in \overline{K}$ such that $u_i \in V_1$ and $v_i \in \overline{V}_1$.

The edge-connectivity of a graph is the minimum number of edges whose removal results in a disconnected or trivial graph.

Corollary. Let K be a cut of the graph G. If the edge-connectivity of $G_K = G - \overline{K}$ is greater than k, then K is k-maximal.

From Definition 1 it is clear that a subset K is maximum with respect to any property if K is $|K|$-maximal. We show the following nontrivial condition for a k-maximal cut to be maximum.

Theorem 5. Let K be a maximal cut of a graph $G = (V,E)$, and let $k = \min\{|K| - |V| + 2, \lceil(|E| - |V|)/2\rceil\}$, where $\lceil x \rceil$ means the least integer $\geq x$. Then K is a maximum cut if K is k-maximal.

(Proof) If K is not a maximum cut, then K is not k'-maximal but (k'-1)-maximal for a certain integer k' such that $k < k'$

$\leq |K|$. By considering the following two cases we show that this leads to a contradiction.

Case 1: $k = |K| - |V| + 2$. By Theorem 3(b) there exists a cutset S_1 of cardinality k' in $G_K = G - \overline{K}$. Since

$$|K - S_1| = |K| - k' \leq |K| - k - 1 = |V| - 3 ,$$

the graph $G_K - S_1 = (K - S_1, V)$ is separated into more than two connected components. Since K is 0-maximal, by Theorem 1 G_K is connected. Therefore S_1 is not a cutset of a connected graph G_K.

Case 2: $k = \lceil (|E| - |V|)/2 \rceil$. By Theorem 3(a) there exists a cutset S of G such that $k' = |K \cap S| < |S - K|$. Therefore

$$|S| \geq 2|K \cap S| + 1 \geq 2k + 3 \geq |E| - |V| + 3.$$

This contradicts the fact that $|S| \leq |E| - |V| + 2$ for any cutset S of a connected graph. Q.E.D.

Note that Theorem 1, 2, 3 and 5 hold true not only for graphs but also for binary matroids [9-10].

4. APPROXIMATION ALGORITHMS

Sahni and Gonzalez [6] have given an approximation algorithm for the maximum cut problem with time complexity $O(|E|)$. On the other hand Kubo et. al. [7] have shown that a maximal cut can be found in time $O(|E|)$, and have given a heuristic algorithm with time complexity $O(|E|^2)$ which works better than the former. However these two algorithms do not guarantee that the solution be k-maximal for $k \geq 1$.

Let $\mathcal{K}^{(k)}(T)$ be a set of cuts determined by a tree T, defined inductively as follows:

$$\mathcal{K}^{(1)}(T) = \{S(e,T) \mid e \in T\} + \{K(T) \oplus S(e,T) \mid e \in T\}$$

and, for $k \geq 2$,

$$\mathcal{K}^{(k)}(T) = \mathcal{K}^{(k-1)}(T) + \{ \bigoplus_{e \in T'} S(e,T) \mid T' \subset T \text{ and } |T'| = k \}$$
$$+ \{K(T) \oplus (\bigoplus_{e \in T'} S(e,T)) \mid T' \subset T \text{ and } |T'| = k \},$$

where $K(T)$ denotes the maximal cut represented by T as $K(T) = \bigoplus_{e \in T} S(e,T)$.

We now present an approximation algorithm, k-APMAXCUT, for the maximum cut problem.

k-APMAXCUT

```
Step 1. $K \leftarrow$ a maximal cut of a given graph G.
Step 2. $T \leftarrow$ a tree $\subseteq K$, of G.
Step 3. If there is a cut $K' \in \mathcal{K}^{(k)}(T)$ such that $|K \oplus K'| >$
        $|K|$, then go to Step 4.  Otherwise stop.
Step 4. $K \leftarrow$ a maximal cut $\subset K \oplus K'$.  Go to Step 2.
```

Theorem 6. A k-maximal cut of a graph can be found in time $O(|V|^k . |E|^2)$ and in space $O(|E|)$ by the algorithm k-APMAXCUT.

The algorithm k-APMAXCUT(t) is an improved version of k-APMAXCUT, obtained by modifying k-APMAXCUT in such a manner that its solution K satisfies $|K \oplus K'| \leq K$ for any cut $K' \in \bigcup_{i=1}^{t} \mathcal{K}^{(k)}(T_i)$, where T_i is a random tree $\subset K$, and t is the number of random trees. Note that the time complexity of k-APMAXCUT(t) is $O(t|V|^k . |E|^2)$, which is equal to the time complexity of k-APMAXCUT if t is a constant independent of the size of a given graph.

The algorithm 1-MAXIMAL, shown in Fig. 1, is designed for the sake of simplicity, so that it works for a given simple graph, but it can easily be modified in such a manner that it works for a given graph with multi edges. The algorithm partly improves Theorem 6.

Theorem 7. A 1-maximal cut of a graph can be found in time $O(|E|^2)$ and in space $O(|E|)$.

(Proof) 1-MAXIMAL finds a bride-edge (i.e., a cutset of cardinality 1) of the graph $G_K = G - \overline{K}$ for a present cut K, and replaces K by a larger one if the bridge-edge satisfies the augmenting condition of Theorem 4, and repeats these operations until there is no bride-edge satisfying the augmenting condition. Theorem 4 guarantees that a cut found by 1-MAXIMAL is 1-maximal. It is clear that the space complexity of 1-MAXIMAL is $O(|E|)$. We shall show that the time complexity is $O(|E|^2)$. First we show that the time required for one execution of BRIDGE(v) is $O(|E|)$. Since the time required for the depth-first search used in BRIDGE(v) is $O(|E|)$, it is sufficient to show that the time required for the management of STACKA and STACKB is $O(|E|)$. BRIDGE(v) pushes each edge in $\overline{K}$ onto STACKA when BRIDGE(v) explores it during the depth-first search. When BRIDGE(v) finds a bridge-edge (v,w),

```
procedure 1-MAXIMAL:
comment A graph G=(V,E) is represented by |V| adjacency lists,
   one for each vertex.  L(v) denotes the list concerning a
   vertex v.  Each edge on the lists is marked whether it
   belongs to K or not.  K is a current cut, and |V|= n;
logical cond; integer COUNT; stack STACKA, STACKB;
begin
   cond:= true;
   while cond = true do
      begin
         for each vertex v∈V do mark v "new";
         COUNT:= 1;
         v:= an arbitrary vertex in V;
         SEARCH(v);
         for each vertex v∈V do mark v "new";
         COUNT:= 1;
         v:= 1;
         make STACKA and STACKB empty;
         cond:= false;
         BRIDGE(v)
      end
end

procedure SEARCH(v):
begin integer array DFNUMBER(n);
   mark v "old";
   DFNUMBER(v):= COUNT;
   COUNT:= COUNT + 1;
   for each vertex w on L(v) do
      if (v,w) is in K then
         if w is marked "new" then
            begin
               SEARCH(w)
            end
end

procedure SAT(w):
begin integer L, H, NUMBER;
   L:= DFNUMBER(w);
   H:= COUNT - 1;
   NUMBER:= 0;
   while L ≤ x ≤ H do
      begin
         pop an edge (x,y) from STACKA;
         if y < L, or H < y then
         comment in this case, the edge (x,y) satisfies the
            property required in the augmenting condition;
            begin
               NUMBER:= NUMBER + 1;
               push the edge (x,y) onto STACKB
            end
      end
   if NUMBER ≤ then
      begin
         pop all edges from STACKB;
         push them onto STACKA
      end
   else cond:= true
end
```

```
procedure BRIDGE(v):
begin integer array LOW(n), FATHER(n);
   mark v "old";
   LOW(v):= COUNT;
   COUNT:= COUNT + 1;
   for each vertex w on L(v) do
      if (v,w) is in K then
         if w is marked "new" then
            begin
               FATHER(w):= v;
               BRIDGE(w);
               if cond = true then goto EXIT
               else
                  if LOW(w) ≥ DFNUMBER(v) then
                  comment the vertex v is a cutpoint;
                     if LOW(w) > DFNUMBER(v) then
                     comment in this case, the edge (v,w) is
                        a bride-edge;
                        begin
                          SAT(w);
                          if cond = true then goto L1
                        end
                  else LOW(v):= MIN(LOW(v), LOW(w))
            end
         else if w is not FATHER(v) then
               LOW(v):= MIN(LOW(v), DFNUMBER(w))
      else push the edge (v,w) onto STACKA
   goto EXIT
L1:while STACKB is not empty do
      begin
         pop an edge (x,y) from top of STACKB;
         add the edge (x,y) to K
      end
   delete the edge (v,w) from K
EXIT:
end
```

Fig. 1 Program of 1-MAXIMAL written in Pidgin ALGOL

BRIDGE(v) calls SAT(w) in order to check whether or not the edge (v,w) (the cut $\{(v,w)\}$) satisfies the augmenting condition, i.e., whether NUMBER ≥ 1 or not, where NUMBER is the cardinality of the edge set $Q = \{(x,y) \mid (x,y) \in \overline{K},\ x \in V_1,\ y \in \overline{V}_1\}$, and V_1 is a subset of V such that $\{(v,w)\} = \{(x,y) \mid (x,y) \in K,\ x \in V_1$ and $y \in \overline{V}_1\}$. The edges (x,y), such that $x \in V_1$, occupy the upper part of STACKA. SAT(w) pops such edges (x,y) from STACKA, and if $y \in \overline{V}_1$, pushes (x,y) onto STACKB. If NUMBER ≤ 1, then the edge in STACKB is returned to STACKA. Otherwise, K is replaced by $(K - \{(v,w)\}) + Q$. Let n_{AI} be the total number of edges which are pushed onto STACKA during one execution of BRIDGE(v), and n_{AO} the total number of edges which are popped from STACKA. n_{BI} and n_{BO}, concerning STACKB, are defined in the same fashion as n_{AI} and n_{AO}. Let n_{be} be the number of the bridge-edges of G_K. Then

$$2|E| \geq 2|\overline{K}| + n_{be} \geq n_{AI} \geq n_{AO} \geq n_{BI} \geq n_{BO} ,$$

because each edge (x,y) in $\overline{K}$ may be pushed onto STACKA twice: as (x,y) and (y,x), and because the total number of edges which are returned to STACKA from STACKB during one execution of BRIDGE(v) is less than or equal to n_{be}. Therefore the time required for the management of STACKA and STACKB during one execution of BRIDGE(v) is $O(|E|)$. Since SEARCH(v) is the simple depth-first search, the execution time is $O(|E|)$. Since $|K|$ increases with each execution of BRIDGE(v), the number of the executions of SEARCH(v) and BRIDGE(v) is less than or equal to $|E|$. Therefore the time complexity of 1-MAXIMAL is $O(|E|^2)$. Q.E.D.

Similarly an algorithm, namely 2-MAXIMAL, is constructed to find a 2-maximal cut of a graph in time $O(|E|^3)$ and in space $O(|E|)$.

5. COMPUTER SIMULATION

In Fig. 2 we show the probability of the solution being a maximum cut when our algorithms, using the solution of the algorithm [7] as the initial one, applied to random

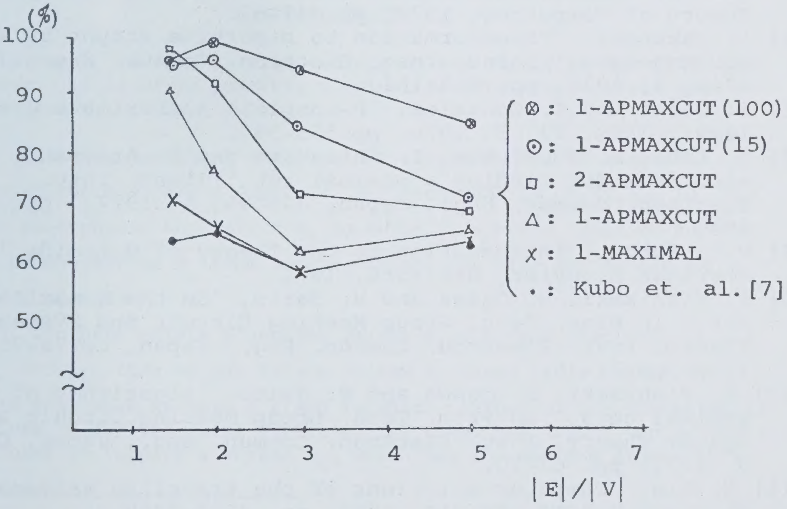


Fig. 2 Probability of approximate solution being maximum.

graphs. We have tested 800 random graphs with 15 vertices and with 24, 30, 45 or 75 edges.

6. CONCLUSION

We introduced the concept "k-maximal" which is similar to "λ-opt" concerning the travelling salesman problem [11], and gave the characterizations of k-maximal cuts of graphs. We showed that the algorithm k-APMAXCUT finds a k-maximal cut in time $O(|V|^k . |E|^2)$, and that the algorithm 1-MAXIMAL finds a 1-maximal cut in time $O(|E|^2)$. The effectiveness of our algolrithms was verified by computer simulations on random graphs. It should be noted that our algorithms are effective especially for sparse graphs, and that 1-APMAXCUT(t) works better than 2-APMAXCUT.

ACNOWLEDGEMENT

The authors would like to thank R.J. Duffin, B. Weide and I. Baybars for their suggestions in writing the manuscript.

REFERENCES

[1] A. Hashimoto and J. Stevens, "Wire routing by optimizing channel assignment with large aperture," Proc. SHARE. ACM. IEEE Design Automation Workshop, 1971, pp. 155-179.
[2] R.M. Karp, "Reducibility among combinatorial problems," in Complexity of Computer Computation, R.E. Miller and J. W. Thatcher, eds, Plenum Press, N.Y., 1972, pp. 85-103.
[3] A.V. Aho, J.E. Hopcroft and J.D. Ullman, "The Design and Analysis of Computer Algorithms," Addison-Wesley, Reading. Mass., Chap. 10, 1974.
[4] M.R. Garey, D.S. Johnson and L. Stockmeyer, "Some simplified NP-complete problems," Proc. 6th Annual ACM Symp. on Theory of Computing, 1974, pp. 47-63.
[5] I. Takanami, "Transformation to bipartite graphs by removing edges," Trans. Inst. Electron. Commun. Eng., Japan, 57-A, 2, 1974, pp. 152-153.
[6] S. Sahni and T. Gonzalez, "P-complete approximate problems," JACM, 23, 3, 1976, pp.555-565.
[7] N. Kubo, S. Tsukiyama, I. Shirakawa and H. Ariyoshi, "An algorithm for finding a maximal cut," Trans. Inst. Electron. Commun. Eng., Japan, J-60-A, 4, 1977, pp. 384-390.
[8] W.T. Tutte, "Introduction to the Theory of Matroids," American Elsevier, New York, 1971.
[9] T. Nishizeki, H. Ogawa and N. Saito, "On the k-maximal cut," in Proc. Tech. Group Meeting Circuit and System Theory, Inst. Electron. Commun. Eng., Japan, CST76-95, 1976, pp.71-76.
[10] T. Nishizeki, H. Ogawa and N. Saito, "Algorithms of k-maximal cuts," in Proc. Tech. Group Meeting Circuit and Syatem Theory, Inst. Electron. Commun. Eng., Japan, CST77-9, 1977, pp. 63-70.
[11] S. Lin, "Computer solutions of the traveling salesman problems," BSTJ, 44, 10, 1965, pp. 2245-2269.

HEURISTIC ALGORITHMS FOR THE SET COVERING PROBLEM AND THEIR BEHAVIOR[1]

K. Y. CHWA AND S. L. HAKIMI
Department of Electrical Engineering and Computer Science
Northwestern University
Evanston, Illinois 60201

ABSTRACT

An $O(n^2)$ heuristic algorithm for the uniform cost covering problem is presented. This algorithm is then used as the basis for an $O(n^3)$ improved algirithm. These algorithms are compared with Bowman and McVey's algorithm on 66 test problems. A worst case analysis is made and it is shown that the ratio, worst solution/optimum solution, is proportional to the log of the maximum degree.

INTRODUCTION

Let $G = (\Gamma,X,Y)$ be a bipartite graph where as usual Γ signifies the adjacency relation between vertices in X and vertices in Y. For $z \in X \cup Y$, by $d(z,G) = |\Gamma z|$ we mean the degree of z in G. We assume $d(y,G) \geq 1$ for all $y \in Y$. The subset $X' \subseteq X$ is called a cover of Y if $\Gamma X' = Y$. Then the problem of finding a cover $X^* \subseteq X$ such that $|X^*| \leq |X'|$ for every possible cover X' of Y, is referred to as the cover problem.

This problem arises in many situations involving combinatorial optimization; some of these are graph theoretic problems [1,2] and others are mathematical programming problems [3,4]. Cover problems also have applications in information theory, switching theory, and fault analysis. General discussion of this problem can be found in [5,6]. This problem has been shown by Karp [3,4] to be an NP-complete problem; therefore to solve large problems, heuristic algorithms must be employed.

Some criteria and several heuristic algorithms have already been proposed [7-9] for solving this problem.

BASIC ALGORITHM

We present, in this section, an elementary algorithm for finding a good cover. It is often possible a priori to eliminate certain vertices in $X \cup Y$ from G.

i) Choosing essential vertices: A vertex $x_i \in X$ is essential if and only if there exists a vertex $y_j \in Y$ such that $\Gamma y_j = \{x_i\}$. In this case, we must choose the vertex x_i to cover y_j since y_j must be covered. We then remove vertex x_i from consideration, and also may remove the vertices $\Gamma x_i \subseteq Y$.

ii) If there exists a pendant vertex $x_i \in X$ such that $d(y_i,G) \geq 2$ where $y_i = \Gamma x_i$, then we may delete vertex x_i since it is redundant.

Note that when we have a graph in which neither of the operation i) and ii) can be applied every vertex of the graph has degree two or more. Our approach is to pick a vertex x_i, which has the maximum degree in X,

---

[1] This work is supported by Air Force Office of Scientific Research, Air Force Systems Command, Grant AFOSR-76-3017.

as a member of a cover. But more careful consideration is needed to choose such a vertex since it provides no information for making a choice if two or more vertices cover the same number of vertices in Y. Suppose $x_i, x_j \in X$ are two candidates, which cover the same number of vertices in Y, i.e. $|\Gamma x_i| = |\Gamma x_j|$. If x_i is chosen as part of our cover, then we can remove x_i and all vertices in Γx_i (similarly for x_j). Thus the number of edges which may be removed after x_i or x_j is chosen, is at least $\sum_{y \in \Gamma x_i} d(y,G)$ or $\sum_{y \in \Gamma x_j} d(y,G)$, respectively. Assume $\sum_{y \in \Gamma x_i} d(y,G) < \sum_{y \in \Gamma x_j} d(y,G)$. Then x_i would be a better choice since the number of edges in the remaining graph $(\Gamma, X-x_i, Y-\Gamma x_i)$ will be more than if x_j were chosen. That is, $(\Gamma, X-x_i, Y-\Gamma x_i)$ is "denser" than $(\Gamma, X-x_j, Y-\Gamma x_j)$ and as a result $(\Gamma, X-x_i, Y-\Gamma x_i)$ has a better chance of having a smaller cover.

Basic Algorithm

Given a bipartite graph $G = (\Gamma, X, Y)$, let $G_1 = (\Gamma, X_1, Y_1)$.

STEP 0: $X_1 \leftarrow X$, $Y_1 \leftarrow Y$, $X^* \leftarrow \emptyset$, $|Y^*| \leftarrow 0$.

STEP 1: Find $Y_1' = \{y \in Y_1 | d(y,G_1) = 1\}$ and $X_1' = \Gamma Y_1'$. $|Y^*| \leftarrow |Y^*| + |\Gamma X_1'|$, $X^* \leftarrow X^* \cup X_1'$, $X_1 \leftarrow X_1'$, $Y_1 \leftarrow Y_1 - \Gamma X_1'$

STEP 2: Find $x_1'' = \{x \in X_1 | d(x,G_1) = 1\}$. If $X_1'' = \emptyset$, go to STEP 4.

STEP 3: For $x \in X_1''$, $X_1'' \leftarrow X_1'' - \{x\}$. If $d(\Gamma x, G_1) \geq 2$, $X_1 \leftarrow X_1 - \{x\}$; otherwise repeat STEP 3 until $x_1'' = \emptyset$, then go to STEP 1.

STEP 4: If $|Y^*| = |Y|$, X^* is a cover set, STOP; otherwise go to next STEP.

STEP 5: Find $\tilde{x}_1 = \{x \in X_1 | d(x,G_1) = \max_{x_i \in X_1} \{d(x_i,G_1)\}\}$ and

$$\hat{x}_1 = \{x \in \tilde{X}_1 | \sum_{y_i \in \Gamma x} d(y_i,G_1) = \min_{x_i \in \tilde{X}_1} \{\sum_{y_j \in \Gamma x_i} d(y_j,G_1\}\}$$

choose $x_1 \in \hat{X}_1$ arbitrarily. $|Y^*| \leftarrow |Y^*| + d(x_1,G_1)$, $X^* \leftarrow X^* \cup \{x_1\}$, $X_1 \leftarrow X_1 - \{x_1\}$.

STEP 6: If $|Y^*| = |Y|$, X^* is a cover set, STOP; otherwise go to STEP 2.

Notice that Y^* indicates a subset of Y which is already covered by X^*. Thus the algorithm will terminate when $|Y^*| = |Y|$.

IMPROVED ALGORITHM

Consider a state diagram as shown in Fig. 1(a). Initial state corresponds to a given graph (problem) $G = (\Gamma, X, Y)$, and a vertical arrow from G_i to G_{i+1} denoted by (G_i, G_{i+1}) indicates that $x_i' \in X$ is the first member of the cover obtained by applying the basic algorithm to the graph G_i, and G_{i+1} is the reduced graph obtained from G_i (until the next member of the cover is found by the basic algorithm). Then we can see that each level corresponds to the cardinality of the current cover. The "dynamics" of the basic algorithm for a given problem can be represented by a state diagram as shown in Fig. 1(a).

Consider the graph $G_i = (\Gamma, X_i, Y_i)$. If $x_i' \in X_i$ is an essential vertex for the cover, we immediately proceed to graph G_{i+1}. If x_i' is not an essential vertex, we call it an initial candidate vertex with respect to G_i. Suppose vertex x_i' is not in any minimal cover set, then we may eliminate it from G_i, which yields the reduced graph $G_i' = (\Gamma, X_i - \{x_i'\}, Y_i)$, without losing any information with regard to an optimum solution. This G_i' can be represented as another state in a state diagram by adding horizongal arrow from G_i into G_i' as in Fig. 1(b). Keeping this in mind, the decision as to whether or not x_i' will be in the final cover is made by comparing the cardinality of the cover sets ℓ_i, ℓ_i' generated by applying basic algorithm to G_i and G_i', respectively, where $\ell_i = \ell_o - i$, and ℓ_o is the cardinality of the cover set generated by the basic algorithm applied to G. If ℓ_i is greater than ℓ_i', we may discard the vertex x_i' since the existence of x_i' does not help in finding a smaller cover set by the basic algorithm, and for $i + 1 \le k \le \ell_i'$. We redefine $G_k = H_k$ where H_k is the reduced graph at the kth level obtained from using the basic algorithm on G_i' (we discard $G_{\ell_i'+1}, \ldots\ldots G_{\ell_i}$). If ℓ_i is less than or equal to ℓ_i', we assume x_i' a member of the final cover. We then proceed to consider graph G_{i+1} in the same fashion. We repeat this procedure until the final state (null graph) is reached.

Improved Algorithm

Given a bipartite graph $G = (\Gamma, X, Y)$, leg $G_1 = (\Gamma, X_1, Y_1)$.

STEP 0: $X_1 \leftarrow X$, $Y_1 \leftarrow Y$, $X_1^* \leftarrow \emptyset$, $X_2^* \leftarrow \emptyset$, $|Y_1^*| \leftarrow 0$, $|Y_2^*| \leftarrow 0$, NUM $\leftarrow$ 1, KR $\leftarrow$ 0, ITER $\leftarrow$ 1.

STEP 1: Find $Y_1' = \{y \in Y_1 | d(y, G_1) = 1\}$ and $X_1' = \Gamma Y_1'$.
$X_{NUM}^* \leftarrow X_{NUM}^* \cup X_1'$, $|Y_{NUM}^*| \leftarrow |Y_{NUM}^*| + |\Gamma X_1'|$, $X_1 \leftarrow X_1 - X_1'$, $Y_1 \leftarrow Y_1 - \Gamma X_1'$

STEP 2: Find $X_1'' = \{x \in X_1' | d(x, G_1) = 1\}$. If $X_1'' = \emptyset$, go to STEP 4.

STEP 3: For $x \in X_1''$, $X_1'' \leftarrow X_1'' - \{x\}$. If $d(\Gamma\{x\}, G_1) \ge 2$, $X_1 \leftarrow X_1 - \{x\}$; otherwise repeat STEP 3 until $X_1'' = \emptyset$, then go to STEP 1.

STEP 4: If $|Y_{NUM}^*| = |Y|$, go to STEP 9; otherwise go to next STEP.

STEP 5: Find $\widetilde{X} = \{x \in X_1 | d(x, G_1) = \max_{x_i \in X_1} \{d(x_i, G_1)\}\}$ and
$\hat{X} = \{x \in \widetilde{X} | \sum_{y_i \in \Gamma x} d(y_i, G_1) = \min_{x_i \in \widetilde{X}} \{\sum_{y_j \in \Gamma x_i} d(y_j, G_1)\}\}$.
Choose $x \in \hat{X}$ arbitrarily.

STEP 6: Find Γx, $X_1 \leftarrow X_1 - \{x\}$. If KR = 1, then $G_2 \leftarrow G_1$, $A_{2x} \leftarrow X_{NUM}^*$, $A_{2y} \leftarrow |Y_{NUM}^*|$, go to Next STEP. If ITER $\neq$ 1, go to Next STEP. If NUM = 1, then $G_2 \leftarrow G_1$, $A_{2x} \leftarrow X_1^*$, $A_{2y} \leftarrow |Y_1^*|$, go to Next STEP; otherwise,

$G_3 \leftarrow G_1$, $A_{3x} \leftarrow X_2^*$, $A_{3y} \leftarrow |Y_2^*|$.

STEP 7: $X_{NUM}^* \leftarrow X_{NUM}^* \cup \{x\}$, $|Y_{NUM}^*| \leftarrow |Y_{NUM}^*| + |\Gamma x|$, $Y_1 \leftarrow Y_1 - \Gamma x$. If KR = 1, $G_4 \leftarrow G_1$, $A_{4x} \leftarrow X_{NUM}^*$, $A_{4y} \leftarrow |Y_{NUM}^*|$, $G_1 \leftarrow G_2$, $X_2^* \leftarrow A_{2x}$, $|Y_2^*| \leftarrow A_{2y}$, set KR $\leftarrow$ 0, ITER $\leftarrow$ 1, go to STEP 1. If ITER $\neq$ 1, go to Next STEP. If NUM = 1, $G_4 \leftarrow G_1$, $A_{4x} \leftarrow X_1^*$, $A_{4y} \leftarrow |Y_1^*|$, go to Next STEP; otherwise, $G_5 \leftarrow G_1$, $A_{5x} \leftarrow X_2^*$, $A_{5y} \leftarrow |Y_2^*|$.

STEP 8: If $|Y_{NUM}^*| = |Y|$, go to Next STEP; otherwise, set ITER $\leftarrow$ 2, go to STEP 2.

STEP 9: If NUM = 1, go to STEP 10(a). If $|X_1^*| \leq |X_2^*|$, go to STEP 10(b); otherwise, go to STEP 10(c).

STEP 10(a): If ITER = 1, then X_1^* is a cover set, STOP; otherwise, $G_1 \leftarrow G_2$, $X_2^* \leftarrow A_{2x}$, $|Y_2^*| \leftarrow A_{2y}$, set NUM $\leftarrow$ 2, ITER $\leftarrow$ 1, go to STEP 1.

STEP 10(b): If ITER = 1, then X_1^* is a cover set, STOP; otherwise, $G_1 \leftarrow G_4$, $X_2^* \leftarrow A_{4x}$, $|Y_2^*| \leftarrow A_{4y}$, set ITER $\leftarrow$ 1, NUM $\leftarrow$ 2, KR $\leftarrow$ 1, go to STEP 2.

STEP 10(c): If ITER = 1, then X_2^* is a cover set, STOP; otherwise, $G_2 \leftarrow G_3$, $A_{2x} \leftarrow A_{3x}$, $A_{2y} \leftarrow A_{3y}$, $G_4 \leftarrow G_5$, $A_{4x} \leftarrow A_{5x}$, $A_{4y} \leftarrow A_{5y}$, $G_1 \leftarrow G_2$, $X_1^* \leftarrow X_2^*$, $X_2^* \leftarrow A_{2x}$, $|Y_2^*| \leftarrow A_{2y}$, ITER $\leftarrow$ 1, go to STEP 1.

Notice that neither of the two algorithms guarantee an irredundant cover, i.e. a cover without any proper subsets which are also covers. But this can be overcome easily. Let a cover set $X^* = \{x_1^*, x_2^*, \ldots, x_\alpha^*\}$ be obtained by either of the two algorithms (preserving the selection order). Consider the subgraph $G_s = (\Gamma, X^*, Y)$. Note that there does not exist x_i^* or $x_j^* \in X^*$ such that $\Gamma x_i \subseteq \Gamma x_j$ with $i \neq j$, and there exists at least a pendant vertex in $\Gamma x_\alpha \subset Y$ and $\Gamma x_{\alpha-1} \subset Y$ where $\alpha = |X^*|$. Therefore if step 1 of both algorithms is applied to $G_s = (\Gamma, X^*, Y)$, the irredundant cover is obtained. In performing the basic algorithm, the procedure is an $O(n^2)$ algorithm in the worst case assuming that $|X|$ and $|Y|$ are of the same order n. In the improved algorithm, we can apply the basic algorithm at most $|X|$ times. Hence the procedure is an $O(n^3)$ algorithm.

WORST CASE ANALYSIS

A simple algorithm which picks a vertex, with maximum degree in X in each iteration, has already been analyzed with respect to the worst case behavior by Johnson [10]; the ratio of the worst solution value, chosen by this algorithm, to the optimal value can grow with the log of maximum degree in X. We will show that this ratio can be slightly reduced. Our approach here is somewhat different than Johnson's and leads to more explicit worst case construction.

We start by stating the following results whose proof is self-evident.

Lemma 1. Given $G = (\Gamma, X, Y)$ with $|Y| = n$ and optimum solution α_o, there exists at least one vertex $x \in X$ such that $d(x,G) \geq \lceil n/\alpha_o \rceil$.

Given n and α_o, we would like to construct a graph $G_o = (\Gamma, X_o, Y_o)$, where $Y_o = \{y_1, y_2, \ldots, y_n\}$, $X_o = X^* \cup X_c$, and $X^* = \{x_1^*, x_2^*, \ldots, x_{\alpha_o}^*\}$ is the optimum cover set, $X_c = \{x_1, x_2, \ldots, x_\alpha\}$ is another cover set, which is the worst possible colution of all solutions obtainable by the basic algorithm.

Construction Algorithm for G_o.

STEP 0: For $1 \le i \le \alpha_o$, assign $\Gamma x_i^* \leftarrow \{y_i, y_{\alpha_o+i}, y_{2\alpha_o+i}, \ldots, y_{j\alpha_o+i}, \ldots\}$.

STEP 1: $d_1 \leftarrow \lceil n/\alpha_o \rceil$, $n_1 \leftarrow n - d_1$, $i \leftarrow 1$.

STEP 2: If $d_i \le 2$, then $\alpha_1 \leftarrow i - 1$, go to STEP 3: otherwise, assign $\Gamma x_i \leftarrow \{y_{n_i+1}, y_{n_i+2}, \ldots, y_{n_i+d_i}\}$, $i \leftarrow i + 1$, $d_i \leftarrow \lceil n_{i-1}/\alpha_o \rceil$, $n_i \leftarrow n_{i-1} - d_i$, repeat STEP 2.

STEP 3: $N_1 \leftarrow n_{i-1} - \alpha_o$, $k \leftarrow \lfloor N_1/3 \rfloor$, $\ell \leftarrow \alpha_o - 3k$. If $k \ge 1$, for $0 \le j \le k - 1$, assign $\Gamma x_{i+j} \leftarrow \{y_{3j+1}, Y_{3j+2}\}$, $\Gamma x_{i+k+j} \leftarrow \{y_{3(j+1)}, y_{\alpha_o+3(j+1)-2}\}$, $\Gamma x_{i+2k+\ell+j} \leftarrow \{y_{3(j+1)}, y_{\alpha_o+3(j+1)-1}\}$, $\Gamma x_{i+3k+\ell+j} \leftarrow \{y_{3(j+1)}, y_{\alpha_o+3(j+1)}\}$. If $\ell \ge 1$, for $0 \le j \le \ell - 1$, assign $x_{i+2k+j} \leftarrow \{y_{3k+1+j}, y_{\alpha_o+3k+1+j}\}$, $G_o = (\Gamma, X^* \cup X_c, Y_o)$, STOP.

Figure 2 shows an example of the worst performance of the basic algorithm for given optimal value, 4, and $|Y_o| = 16$. In the other algorithms [7-9], the worst solutions of the same example (Fig. 2) are not better than the basic algorithm. Consider a graph $G_o = (\Gamma, X_o, Y_o)$ which is obtained by the construction algorithm. Let us define $G_i = (\Gamma, X_i, Y_i) \supset G_{i+1} = (\Gamma, X_{i+1}, Y_{i+1})$ for $0 \le i \le \alpha - 1$, where $X_{i+1} = X_i - \{x_{i+1}\}$, $Y_{i+1} = Y_i - \Gamma\{x_{i+1}\}$ and X_{i+1} is defined in the construction algorithm. Then we can observe than the graph G_o has the following properties:

$$\max_{x^* \in X^*} \{d(x^*, G_i)\} \le \min_{x^* \in X^*} \{d(x^*, G_i)\} + 1$$

$$d_{i+1} \triangleq d(x_{i+1}, G_i) = \max_{x^* \in X^*} \{d(x^*, G_i)\} \text{ for } 0 \le i \le \alpha - 1 \ .$$

Define n_i, the number of vertices in Y_o after $x_i \in X_c$ is chosen, we have:

$$n_1 = n - d_1 \ , \quad n_i = n_{i-1} - d_i \qquad \text{for } 1 \le i \le \alpha \ .$$

Notice that α_1 (in the construction algorithm) indicates the number of vertices in X_C which are assigned to $\Gamma\{x_i\}$ in STEP 2, and $|\Gamma\{x_i\}| \ge 3$ for $1 \le i \le \alpha_1$ while $|\Gamma\{x_i\}| = 2$ for $\alpha_1 + 1 \le i \le \alpha$. That is, α_1 is the number of vertices which cover more than two vertices when they are chosen. We can see in the construction algorithm that $n_{\alpha_1} \le 2\alpha_o < n_{\alpha_1 - 1} \le 2\alpha_o + 3$ and $\alpha - \alpha_1 \le 4/3\ \alpha_o$.

Lemma 2. Given $|Y_o| = n$ and optimum solution value α_o, the worst solution for the graph $G_o = (\Gamma, X^* \cup X_c, Y_o)$ by the basic algorithm is not less than the corresponding solution for any other construction.

Proof. Let $G_o' = (\Gamma', X_o', Y_o)$ and $|Y_o| = n$ be a graph which is constructed by any other method, with an optimum cover of cardinality α_o, and $X_c' \subset X_o'$ be the worst possible cover set which is found by the basic algorithm. Let d_i' denote the number of vertices in Y_o covered by $x_i' \in X_c'$ when x_i' is chosen (by the basic algorithm), and assume that the x_i's are rearranged such that $d_i' \geq d_{i+1}'$. Let n_i' denote the number of uncovered vertices in Y_o after $x_i' \in X_c'$ is chosen. d_i' and n_i' obey similar relations to d_i and n_i respectively. We now claim that $n_i \geq n_i'$ for $1 \leq i \leq \alpha$. For $i = 1$, $n_1 - n_i' = (n-d_1) - (n-d_1') = d_1' - d_1$. Since $d_1' \geq \lceil n/\alpha_o \rceil$ (by Lemma 1) and $d_1 = \lceil n/\alpha_o \rceil$, $d_1' - d_1 \geq 0$, hence $n_1 - n_1' \geq 0$. Suppose the assumption is true for all $i \leq k-1$. Let $n_{k-1} - n_{k-1}' = P\alpha_o + q$ where P,q are non-negative integers and $0 \leq q < \alpha_o$.

Since $d_k = \lceil n_k - 1/\alpha_o \rceil$ and $d_k' \geq \lceil n_{k-1}'/\alpha_o \rceil$,

$$\begin{aligned} n_k - n_k' &= (n_{k-1} - d_k) - (n_{k-1}' - d_k') \\ &= P(\alpha_o - 1) + q + \lceil n_{k-1}'/\alpha_o \rceil - \lceil (n_{k-1}' + q)/\alpha_o \rceil \geq 0. \end{aligned}$$

Therefore $n_\alpha = 0 \geq n_\alpha'$. Hence, the Lemma is true.

Theorem. Let G be an arbitrary graph and X^* be an optimum solution with $|X^*| = \alpha_o$. Let X_c be the worst solution obtained by the basic algorithm. Then,

$$\frac{|X_c|}{\alpha_o} < 1/\alpha_o + 4/3 + \ln(n/2\alpha_o)/\alpha_o \ln(1 + 1/(\alpha_o - 1)).$$

Proof. Let $\gamma = 1/\alpha_o$ and let for each positive number $u, n_u'' = n(1-\gamma)^u$. We will prove (by induction) that, $n_i'' \geq n_i$ for all integral values of i.

For $i = 1$, $n_1'' - n_1 = n(1-\gamma) - (n - \lceil n\gamma \rceil) \geq 0$.

Suppose the assumption is true for all $i \leq k-1$.

Let $n_{k-1}'' - n_{k-1} = P\alpha_o + q + t$ where P and q are non-negative integers and t is a real number such that $0 \leq q < \alpha_o$, $0 \leq t < 1$.

$$\begin{aligned} n_k'' - n_k &= n(1-\gamma)^k - n_{k-1} - (\lceil n_{k-1} \cdot \gamma \rceil) \\ &= n_{k-1}''(1-\gamma) - (n_{k-1} - \lceil \gamma \cdot n_{k-1} \rceil) \\ &= (P\alpha_o + q + t)(1-\gamma) + \lceil \gamma \cdot n_{k-1} \rceil - \gamma n_{k-1} \geq 0 \end{aligned}$$

Therefore, $n_i'' \geq n_i$.

Now let α' be defined by $n_{\alpha'}'' = 2\alpha_o$, i.e. $\alpha' = \ln(n/2\alpha_0)/\ln(1+1/(\alpha_o - 1))$. Note that α' is not an integer $n_{\alpha-1} > n_{\alpha'}$. Since $n_{\alpha_1 - 1}'' \geq n_{\alpha_1 - 1} >$ $> 2\alpha_o = n_{\alpha'}''$, and $\alpha - \alpha_1 \leq 4/3\alpha_o$, $\alpha \leq \alpha_1 + 4/3\ \alpha_o < \alpha' + 4/3\ \alpha_o + 1$.

So the ratio α/α_o becomes

$$\alpha/\alpha_o < 4/3 + \ln(n/2\alpha_o)/\alpha_o \ln(1 + 1/(\alpha_o - 1)) + 1/\alpha_o$$

Hence the theorem is proved.

To simplify this result, $\alpha_o \ln(1 + 1/(\alpha_o - 1)) = \alpha_o(1/(\alpha_o - 1) - \frac{1}{2(\alpha_o - 1)^2}$

$$+ \frac{1}{3(\alpha_o - 1)^3} - \ldots) \geq \alpha_o\left(\frac{1}{\alpha_o - 1} - \frac{1}{2(\alpha_o - 1)^2}\right) \geq 1$$

In particular, since $\max\{d(x,G)\} \geq n/\alpha_o$ and $\alpha_o \geq 2$,

$$\alpha/\alpha_o < 1/\alpha_o + 4/3 + \ln(n/2\alpha_o) \leq 1/\alpha_o + 4/3 + \min\left\{\ln \frac{d_{max}}{2}, \ln(n/4)\right\}$$

where $d_{max} = \max\{d(x,G)\}$.

In the proof of the theorem we have not used the decision rule, that is: choose a vertex x, as in STEP 5 of the basic algorithm, in finding α'. However, the authors feel that this decision rule has an important role to play in keeping the solution near the optimum solution. Consider the edge covering problem which can be solved in polynomial time ($O(|V|^3)$) [11,12]. The edge covering problem is equivalent to our model, restricted so that the degree of every vertex in X is exactly two. Therefore, the maximum degree in X becomes meaningless. Given G = (V,E), if we apply the basic algorithm to the edge covering problem, we can interpret our decision rule as follows: pick an edge e_k such that the sum of the degrees of the two vertices incident at e_k is minimum. Let us assume optimum solution is α_o. Without decision rule, the worst case ratio becomes 3/2. However, applying the basic algorithm, the worst case ratio becomes 5/4; this is because the basic algorithm always finds optimum solution when $\alpha_o < 4$. Fig. 3 shows the worst construction when $\alpha_o = 4$.

EXPERIMENTAL RESULTS

In order to compare the results of both algorithms, a series of experiments has been designed. Since it is nearly impossible to obtain the real optimum solutions, we programmed Bowman and McVey's algorithm and compared its results to ours. The program was written in FORTRAN and run on a CDC 6400.

Let $|X| = n$ and $|Y| = m$, the successor list is produced by generating a random number sequence between 1 and n. In the experiments, m and n are equal to 100. The problems generated are divided into three groups.

In the first group, all the degrees of the vertex in X are equal to 5 and all the degrees of the vertices in Y are larger than or equal to 2. This process is then repeated for vertex degrees in X of values 6,7,8,9,10. Three different problems for each case are produced. Thus the first group contains 18 problems.

The second group is similar to the first group with X and Y interchanged - another 18 problems.

In the third group, another random number sequence is produced for the degree of the vertices in X varying from 2 to 8. This process is repeated for degrees varying from 2 to 10, 2 to 13, 2 to 15, and 2 to 20. Three different problems for each case are generated, making 15 problems. X and Y are then interchanged and the whole procedure is repeated. This gives 30 problems for the third group. Altogether there are 66 problems.

Let α_{bm}, α_b, and α_i denote values obtained from Bowman and McVey's, the basic, and the improved algorithm, respectively. For $j, k \in \{bm, b, i\}$, we define the relative difference of α_j and α_k by $D_k^j = (\alpha_j - \alpha_k)/\alpha_k$. We now plot the percentile curve with percentage difference ($D_i^{bm} \times 100$, $D_i^b \times 100$) not exceeding γ as in Fig. 4.

In Fig. 4, the interesting feature is that only 2 solutions by Bowman and McVey's algorithm out of 66 problems were better than the solution by the improved algorithm, 10 were the same while the rest, 54 cases, were worse. The basic algorithm had 25 the same, and the rest worse when compared with the improved algorithm. Also Bowman and McVey's algorithm has 17 cases with solutions worse than 10% of the solutions of the improved algorithm, and 44 cases with solutions worse than 5%; while the basic algorithm has corresponding figures of 4 and 28 cases, respectively.

In order to see the behavior of the algorithms by groups and densities, the results of each group of experiments have been tabulated as in Tables 1 through 3 by taking mean for each of the densities over the three cases. From these Tables we may say that the improved algorithm dominates Bowman and McVey's algorithm in every case.

The computation time for Bowman and McVey's algorithm lies between 1.073 and 2.263 seconds with mean value 1.485, and for the basic algorithm between 1.223 and 2.378 with mean value 1.635 seconds and for the improved algorithm between 2.171 and 9.391 seconds with mean value 6.285.

REFERENCES

1. J. F. Desler, "Degree Contrained Subgraphs, Covers, Codes and K-Graphs," Ph.D. dissertation, Northwestern Univ., 1969.

2. S. L. Hakimi, "Steiner's Problem in Graphs and Its Implications," NETWORKS 1, 1971, pp. 113-133.

3. R. M. Karp, "Reducibility Among Combinatorial Problems," Complexity of Computer Computations, R. E. Miller and J. W. Thatcher, eds., Plenum Press, New York, 1972, pp. 85-104.

4. R. M. Karp, "on The Computational Complexity of Combinatorial Problems," NETWORKS 5, 1975, pp. 45-68.

5. R. S. Garfinkel and G. L. Nemhauser, "Optimal Set Covering: A Survey," in Perspectives in Optimization: A Collection of Expository Articles (ed. A. M. Geoffrion), Addison-Wesley, Reading, Mass., 1972.

6. H. M. Salkin and J. Saha, "Set Covering: Uses, Algorithms, Results," Tech. Memo No. 272, 1973, Case Western Reserve Univ.

7. R. Roth, "Computer Solutions to Minimum-Cover Problems," Operations Research, 1968, pp. 455-465.

8. R. M. Bowman and E. S. McVey, "A Method for the Fast Approximate Solution of Large Prime Implicant Charts," IEEE Trans. on Computers C-19, Feb. 1970, pp. 169-173.

9. R. P. Batni, J. D. Russell and C. R. Kime, "An Efficient Algorithm for finding an Irredundant Set Cover," J. of ACM Vol. 21 No. 3, Jul., 1974, pp. 351-355.

10. D. S. Johnson, "Approximation Algorithms for Combinatorial Problems," J. Comput. Systems Sci., Sept., 1974, pp. 256-278.

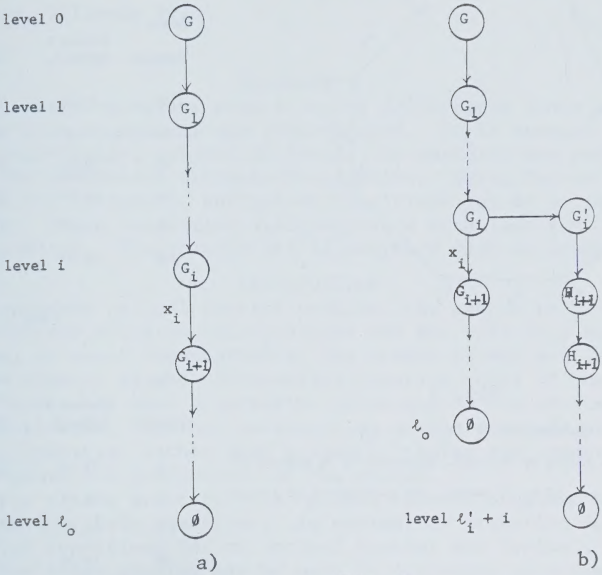

level 0
level 1
level i
level ℓ₀
G
G₁
Gᵢ
xᵢ
Gᵢ₊₁
Ø
a)
G
G₁
Gᵢ
G'ᵢ
xᵢ
Gᵢ₊₁
Hᵢ₊₁
Hᵢ₊₁
ℓ₀
Ø
level ℓ'ᵢ + i
Ø
b)

Fig. 1

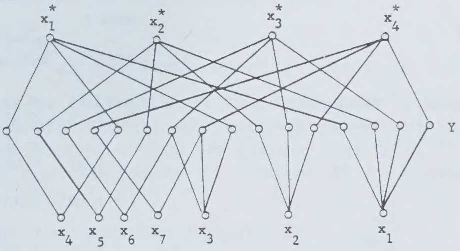

x₁*
x₂*
x₃*
x₄*
Y
x₄
x₅
x₆
x₇
x₃
x₂
x₁

Fig. 2

Optimum Cover Set: $\{x_1^*, x_2^*, x_3^*, x_4^*\}$

Worst Solution of the Basic Algorithm: $\{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$

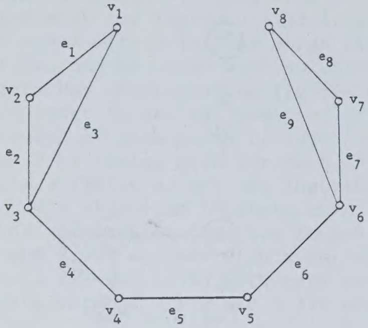


Fig. 3

Optimum Cover: $\{e_1, e_4, e_6, e_8\}$

The first possible choice by the basic algorithm is an element of the set $\{e_1, e_5, e_8\}$. If e_5 is chosen, the cardinality of a solution set becomes 5.

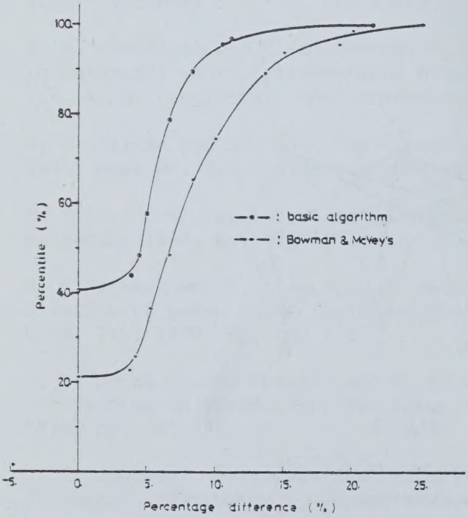


Fig. 4

Group 1

Density	Bowman & McVey's	Basic	Improved
5%	27.66	27.33	26.33
6%	23.66	23.00	22.00
7%	22.33	21.66	20.33
8%	19.00	19.33	18.33
9%	18.00	17.33	16.33
10%	16.00	15.66	14.66

Table 1.

Group 2

Density	Bowman & McVey's	Basic	Improved
5%	23.33	21.00	20.33
6%	19.66	19.00	18.00
7%	18.66	16.66	16.00
8%	16.66	15.00	15.00
9%	14.00	13.33	13.00
10%	14.33	12.66	12.66

Table 2.

Group 3

Density	Bowman & McVey's	Basic	Improved
4.97 - 5.17%	25.33 23.33	25.33 22.66	24.33 22.33
5.47 - 6.29%	21.66 22.33	21.66 21.33	21.00 20.66
7.23 - 7.55%	17.00 19.66	16.66 19.33	16.33 19.33
7.83 - 8.38%	14.33 21.66	15.33 21.66	13.66 21.00
9.81 - 12.09%	12.00 15.66	12.00 16.33	11.00 15.33

Table 3.

MULTICRITERIA OPTIMIZATION WITH UNCERTAINTY IN THE DYNAMICS[1]

W. E. Schmitendorf
Northwestern University
Evanston, Illinois 60201

ABSTRACT

Optimal control problems with a vector performance index and uncertainty in the state equations are investigated. It is assumed that nature chooses the uncertainty, subject to bounds, to maximize the performance index which the controller attempts to minimize. Using Pareto optimality as the optimality criterion, sufficient conditions for an optimal solution are presented. These conditions also suggest a technique for determining the optimal control. The results are illustrated with an example.

1. INTRODUCTION

In the standard optimal control problem, the system to be controlled is modeled by ordinary differential equations and the cost is a scalar. In many problems, an exact description of the system is not available and disturbances are present in the differential equation model of the system. These disturbances may have a corrosive influence on the system performance if they are neglected. Another modification of the standard problem is that multiple criteria, rather than a single, scalar performance index, may be used to measure the performance of the system.

This paper treats multicriteria optimal control problems where there is uncertainty in the state equations. An optimality criterion is defined and then sufficient conditions for an optimal control are derived. In addition, it is shown how these results may be used to determine an optimal solution.

2. PROBLEM FORMULATION

Consider a control system described by the differential equations

$$\dot{x}(t) = f(x(t),u(t),v(t)) \tag{1}$$

where $x(t) \in R^n$ is the state, $u(t) \in R^{m_1}$ is the control and $v(t) \in R^{m_2}$ is the disturbance. We assume that $f(\cdot,\cdot,\cdot)$ is a C^1 function. The time interval $[t_o,t_f]$ is specified as is the initial state.

$$x(t_o) = x_o \tag{2}$$

The disturbance $v(\cdot)$ is not known exactly; the only information available about $v(\cdot)$ is that it belongs to a known set $V \subset R^{m_2}$. Rather than modeling the system in a stochastic manner, we assume that nature chooses the disturbance in an antagonistic manner. Thus we assume that $v(\cdot)$ is chosen to maximize the criteria that the controller is attempting to minimize. This is a worst case design philosophy. Let

$$M_1 = \left\{u(\cdot) : u(\cdot) \in \mathcal{U} \text{ and } u(t) \in U,\ t \in [t_o,t_f]\right\}$$

where $\mathcal{U}$ is the set of piecewise continues functions from $[t_o,t_f] \to R^{m_1}$ and U is a given subset of R^{m_1}. Similarly, let

$$M_2 = \left\{v(\cdot) : v(\cdot) \in \mathcal{V} \text{ and } v(t) \in V,\ t \in [t_o,t_f]\right\}$$

where $\mathcal{V}$ is the set of piecewise continuous functions from $[t_o,t_f] \to R^{m_2}$. For $u(\cdot) \in M_1$ and $v(\cdot) \in M_2$, the solution of (1), (2) will be denoted by $x(\cdot)$. The pair $[u(\cdot),v(\cdot)]$ is playable if it generates a solution of (1), (2)

---

[1] This research was supported by AFOSR under Grant AFOSR-76-2923.

such that $x(t_f)\epsilon\theta$ where the target set θ is a given set in R^n.

For $u(\cdot)\epsilon M_1$, $N(u(\cdot))$ is the set of all $v(\cdot)\epsilon M_2$ such that $[u(\cdot),v(\cdot)]$ is playable. A control $u(\cdot)$ is admissible if $u(\cdot)\epsilon M_1$ and $N(u(\cdot))\neq \emptyset$.

For a playable pair $[u(\cdot),v(\cdot)]$, the cost is a vector with two components,

$$J_i(u(\cdot),v(\cdot);x(\cdot)) = g_i(x(t_f)) + \int_{t_o}^{t_f} L_i(x(t),u(t),v(t))dt, \quad i=1,2 \tag{3}$$

Here $x(\cdot)$ denotes the trajectory corresponding to $u(\cdot),v(\cdot)$ starting at (x_o,t_o). (The results presented below can easily be extended to the case where the cost is an K vector. For simplicity of presentation, we concentrate on the case K=2.) Our optimality criterion is an extension of the concept of Pareto optimality for problems where $v(\cdot)$ is not present.

Definition A control $u^*(\cdot)$ is Pareto optimal if, and only if, it is admissible and for all admissible $u(\cdot)$ either

$$\sup_{v(\cdot)\epsilon N(u^*(\cdot))} J_i(u^*(\cdot),v(\cdot);x(\cdot)) = \sup_{v(\cdot)\epsilon N(u(\cdot))} J_i(u(\cdot),v(\cdot);x(\cdot)), \quad i=1,2$$

or, for at least one $i\ \epsilon\{1,2\}$,

$$\sup_{v(\cdot)\epsilon N(u^*(\cdot))} J_i(u^*(\cdot),v(\cdot);x(\cdot)) < \sup_{v(\cdot)\epsilon N(u(\cdot))} J_i(u(\cdot),v(\cdot);x(\cdot))$$

When the cost has only one component, the definition becomes that of a minmax solution while, if there is no disturbance, the definition is the usual definition of Pareto optimality. If an admissible control, $u_1(\cdot)$, is not Pareto optimal, then there is another admissible control, $u_2(\cdot)$, such that

$$\sup_{v(\cdot)\epsilon N(u_2(\cdot))} J_i(u_2(\cdot),v(\cdot);x(\cdot)) \leq \sup_{v(\cdot)\epsilon N(u_1(\cdot))} J_i(u_1(\cdot),v(\cdot);x(\cdot)), \quad i=1,2$$

with the strict inequality holding for at least one i. Thus, if an admissible control is not Pareto optimal, there is another admissible control which guarantees that no component of the cost vector increases while at least one decreases.

Static versions of this problem are treated in [1-3]. In [4], a many player differential game problem with coalitions was studied and that problem can be interpreted as an optimal control problem with disturbances. There, however, in deriving sufficient conditions, it was assumed that a related two-player zero-sum game has a saddle point solution. We do not make that assumption here.

In the next section two sufficient conditions for Pareto optimal control are presented. These conditions are not directly working conditions, but only a step in that direction. In Sec. 4, we use the second of these conditions to obtain other conditions which may be used for determing Pareto optimal controls.

3. PRELIMINARY RESULTS

Sufficient conditions for Pareto optimality will be obtained in terms of the following two-player zero-sum game.

$$\hat{J}(u(\cdot),v_1(\cdot),v_2(\cdot);x_1(\cdot),x_2(\cdot)) = \alpha_1 J_1(u(\cdot),v_1(\cdot);x_1(\cdot)) + \alpha_2 J_2(u(\cdot),v_2(\cdot);x_2(\cdot)) \tag{4}$$

$$\dot{x}_1(t) = f(x_1(t),u(t),v_1(t)), \qquad x_1(t_o) = x_o \tag{5}$$

$$\dot{x}_2(t) = f(x_2(t),u(t),v_2(t)), \qquad x_2(t_o) = x_o \tag{6}$$

Here $x_1(t),x_2(t)\epsilon R^n, v_1(\cdot),v_2(\cdot)\epsilon M_2$ and admissible $u(\cdot)$ are defined as in Sec. 2. One player controls $u(\cdot)$ while the other controls $v_1(\cdot)$ and $v_2(\cdot)$. Note that $x_i(\cdot)$ and $v_i(\cdot)$ only enter $\hat{J}$ through J_i while $u(\cdot)$enters through $J_1$and J_2. The time interval $[t_o,t_f]$is fixed and we require $x_1(t_f), x_2(t_f)\epsilon\theta$.

The controls $[u^*(\cdot),v_1^*(\cdot),v_2^*(\cdot)]$ are a saddle point solution for this game if they satisfy

$$\hat{J}(u^*(\cdot),v_1(\cdot),v_2(\cdot);x_1(\cdot),x_2(\cdot)) \le \hat{J}(u^*(\cdot),v_1^*(\cdot),v_2^*(\cdot);x_1^*(\cdot),x_2^*(\cdot))$$
$$\le \hat{J}(u(\cdot),v_1^*(\cdot),v_2^*(\cdot);x_1(\cdot),x_2(\cdot)) \tag{7}$$

for $v_1(\cdot),v_2(\cdot)\epsilon N(u^*(\cdot))$and for all $u(\cdot)$that are playable against $v_1^*(\cdot)$ and $v_2^*(\cdot)$. In Theorem 3.1,we assume all $u(\cdot)\epsilon M_1$ are playable against$(v_1^*(\cdot),v_2^*(\cdot))$.

Theorem 3.1 If $[u^*(\cdot),v_1^*(\cdot),v_2^*(\cdot)]$ is a saddle point solution to the differential game (4)-(6) for some $\alpha_1,\alpha_2 > 0$, $\alpha_1+\alpha_2= 1$, then $u^*(\cdot)$ is a Pareto optimal control for the problem (1)-(3).

Proof. The saddle point inequalities (7) imply

$$\hat{J}(u^*(\cdot),v_1^*(\cdot),v_2^*(\cdot);x_1^*(\cdot),x_2^*(\cdot)) = \sup_{v_1(\cdot),v_2(\cdot)\epsilon N(u^*(\cdot))} \hat{J}(u^*(\cdot),v_1(\cdot),v_2(\cdot);x_1(\cdot),x_2(\cdot))$$

and that for any admissible $u(\cdot)$

$$\hat{J}(u^*(\cdot),v_1^*(\cdot),v_2^*(\cdot);x_1^*(\cdot),x_2^*(\cdot)) \le \sup_{v_1(\cdot),v_2(\cdot)\epsilon N(u(\cdot))} \hat{J}(u(\cdot),v_1(\cdot),v_2(\cdot);x_1(\cdot),x_2(\cdot))$$

Thus

$$\sup_{v_1(\cdot),v_2(\cdot)\epsilon N(u^*(\cdot))} \hat{J}(u^*(\cdot),v_1(\cdot),v_2(\cdot);x_1(\cdot),x_2(\cdot))$$
$$\le \sup_{v_1(\cdot),v_2(\cdot)\epsilon N(u(\cdot))} \hat{J}(u(\cdot),v_1(\cdot),v_2(\cdot);x_1(\cdot),x_2(\cdot)) \tag{8}$$

for all admissible $u(\cdot)$. Since $v_i(\cdot)$ and $x_i(\cdot)$ only affect J_i

$$\sup_{v_1(\cdot),v_2(\cdot)\epsilon N(u(\cdot))} \hat{J}(u(\cdot),v_1(\cdot),v_2(\cdot);x_1(\cdot),x_2(\cdot)) = \sum_{i=1}^{2} \alpha_i \sup_{v_i(\cdot)\epsilon N(u(\cdot))} J_i(u(\cdot),v_i(\cdot);x_i(\cdot)) \tag{9}$$

From (8), (9) it follows that

$$\sum_{i=1}^{2} \alpha_i \sup_{v_i(\cdot)\epsilon N(u^*(\cdot))} J_i(u^*(\cdot),v_i(\cdot);x_i(\cdot)) \le \sum_{i=1}^{2} \sup_{v_i(\cdot)\epsilon N(u(\cdot))} J_i(u(\cdot),v_i(\cdot);x_i(\cdot)) \tag{10}$$

Since $\alpha_1,\alpha_2 > 0$, (10) implies that $u^*(\cdot)$ is Pareto optimal. □

From this theorem, it is seen that Pareto optimal solutions can be found by determining open loop saddle point solutions for the two-player zero-sum game (4)-(6). As α_1 and α_2 vary, different saddle point solutions may be obtained. Each of these solutions has the Pareto optimal property. Techniques for finding open loop saddle point solutions can be found in [4-8]. Unfortunately, many of the differential games (4)-(6) which arise from problems with a disturbance (1)-(3) do not have saddle point solutions. They do, however, have minmax solutions. An admissible control $u^*(\cdot)$ is a minmax solution to the game (4)-(6) if it satisfies for all admissible $u(\cdot)$

$$\sup_{v_1(\cdot),v_2(\cdot)\in N(u(\cdot)} \hat{J}(u^*(\cdot),v_1(\cdot),v_2(\cdot);x_1(\cdot),x_2(\cdot))$$

$$\leq \sup_{v_1(\cdot),v_2(\cdot)\in N(u(\cdot))} \hat{J}(u(\cdot),v_1(\cdot),v_2(\cdot);x_1(\cdot),x_2(\cdot))$$

Theorem 3.2 If $u^*(\cdot)$ is a minmax solution to the game (4)-(6) for some $\alpha_1,\alpha_2 > 0$, $\alpha_1+\alpha_2= 1$, then $u^*(\cdot)$ is a Pareto optimal control for the problem (1)-(3).

Proof. Since $u^*(\cdot)$ is a minmax solution, it satisfies (8). The proof then follows in exactly the same manner as the proof of Theorem 3.1. ☐

Theorems similar to Theorem 3.1 and Theorem 3.2 were obtained for problems involving coalitions in a differential game [4]. There, use was made of Theorem 3.1 to find coalitive Pareto optimal controls. Here, we shall concentrate on Theorem 3.2 and use it, in conjunction with a sufficient condition for minmax control [9], to obtain sufficient conditions for a Pareto optimal solution to the problem with disturbances (1)-(3). Also a possible technique for determining Pareto optimal solutions, based on these sufficient conditions, will be discussed.

4. MAIN RESULT

In [8,9] there are sufficient conditions for a control to be a minmax control. Based on these results, as well as Theorem 3.2, we obtain sufficient conditions for a Pareto optimal control for the original problem (1)-(3). First we need some definitions. For i=1,2,the set $\mathcal{L}_i(u(\cdot))$ consists of the admissible disturbances that are playable against $u(\cdot)$ and maximize J_i.

$$\mathcal{L}_i(u(\cdot)) = \Big\{\hat{v}(\cdot):\hat{v}(\cdot)\in N(u(\cdot)) \text{ and } J_i(u(\cdot),\hat{v}(\cdot);x(\cdot)) = \sup_{v(\cdot)\in N(u(\cdot))} J_i(u(\cdot),v(\cdot);x(\cdot))\Big\} \qquad i=1,2$$

Let $w(\cdot,\cdot)$ be a function from $R^n \times R^{m_1} \to R^{m_2}$ and, for $u(\cdot)\in M_1$, $q_w(\cdot)$ is the function defined by $q_w(t) = w(x(t),u(t))$ where $x(\cdot)$ is the solution of $\dot{x}(t) = f(x(t),u(t),w(x(t),u(t)),x(t_o) = x_o$. The functions $w(\cdot,\cdot)$ will be called response functions. The set of admissible response function is

$$\mathfrak{W} = \Big\{w(\cdot,\cdot):q_w(\cdot)\in N(u(\cdot)) \text{ for all admissible } u(\cdot)\Big\}$$

Let γ be a positive integer, $y^i(\cdot,\cdot)\in\mathfrak{W}$ i=1,..,γ, $z^i(\cdot,\cdot)\in\mathfrak{W}$, i=1,..,$\gamma$. Consider an optimal control problem with cost

$$K(u(\cdot),y^i(\cdot,\cdot),z^i(\cdot,\cdot);x_1^i(\cdot),x_2^i(\cdot))$$

$$= \sum_{i=1}^{\gamma} \mu_i \Big\{\alpha_1 J_1(u(\cdot),y^i(\cdot,\cdot);x_1^i(\cdot)) + \alpha_2 J_2(u(\cdot),z^i(\cdot,\cdot);x_2^i(\cdot))\Big\} \tag{11}$$

and state equations

$$\dot{x}_1^i(t) = f[x_1^i(t),u(t),y^i(x_1^i(t),u(t))], \quad x_1^i(t_o) = x_o \quad i=1,..,\gamma \tag{12a}$$

$$\dot{x}_2^i(t) = f[x_2^i(t),u(t),z^i(x_2^i(t),u(t))], \quad x_2^i(t_o) = x_o \quad i=1,..,\gamma \tag{12b}$$

The terminal conditions are $x_1^i(t_f)$, $x_2^i(t_f) \epsilon \theta$, $i=1,..,\gamma$.

In the following, we shall require $\alpha_1 > 0$, $\alpha_2 > 0$. Letting $\rho = \alpha_2/\alpha_1$, (11) can be rewritten as

$$K(u(\cdot),y^i(\cdot,\cdot),z^i(\cdot,\cdot);x_1^i(\cdot),x_2^i(\cdot))$$

$$= \sum_{i=1}^{\gamma} \mu_i J_1(u(\cdot),y^i(\cdot,\cdot);x_1^i(\cdot)) + \rho \sum_{i=1}^{\gamma} \mu_i J_2(u(\cdot),z^i(\cdot,\cdot);x_2^i(\cdot)) \qquad (11')$$

We now present a sufficient condition for a control $u^*(\cdot)$ to be a min-max control for the problem (4)-(6) and thus, from Theorem 3.2, a Pareto optimal control for the problem (1)-(3).

Theorem 4.1 Let $u^*(\cdot)$ be admissible. If there exists

i) a positive integer γ

ii) scalars $\mu_i > 0$, $i=1,..,\gamma$, $\sum_{i=1}^{\gamma} \mu_i = 1$ and a scalar $\rho > 0$

iii) admissible response $y^i(\cdot,\cdot),z^i(\cdot,\cdot) \epsilon \mathcal{W}$, $i=1,..,\gamma$

such that

a) $v_1^{i*}(\cdot) \epsilon \mathcal{L}_1(u^*(\cdot))$, $v_2^{i*}(\cdot) \epsilon \mathcal{L}_2(u^*(\cdot))$, $i=1,..,\gamma$

where $v_1^{i*}(t) = y^i(x_1^{i*}(t),u^*(t))$, $v_2^{i*}(t) = z^i(x_2^{i*}(t),u^*(t))$ and $x_1^{i*}(\cdot),x_2^{i*}(\cdot)$ are solutions of

$$x_1^i(t) = f[x_1^i(t),u^*(t),y^i(x_1^i(t),u^*(t))],\ x_1^i(t_o) = x_o \qquad i=1,..,\gamma$$

$$x_2^i(t) = f[x_i^2(t),u^*(t),z^i(x_2^i(t),u^*(t))],\ x_2^i(t_o) = x_o \qquad i=1,..,\gamma$$

b) $u^*(\cdot)$ is an optimal control for the problem $P(\gamma,\rho,\mu_i,y^i(\cdot,\cdot),z^i(\cdot,\cdot))$ then $u^*(\cdot)$ is a Pareto optimal control for the problem (1)-(3).

Proof. Consider any admissible $\hat{u}(\cdot)$. Since $u^*(\cdot)$ is a solution of the problem $P(\gamma,\rho,\mu_i,y^i(\cdot,\cdot)),\ z^i(\cdot,\cdot))$

$$\sum_{i=1}^{\gamma} \mu_i J_1(u^*(\cdot),v_1^{i*}(\cdot);\ x_1^{i*}(\cdot)) + \rho \sum_{i=1}^{\gamma} \mu_i J_2(u^*(\cdot),v_2^{i*}(\cdot));\ x_2^{i*}(\cdot))$$

$$\leq \sum_{i=1}^{\gamma} \mu_i J_1(\hat{u}(\cdot);q_{\hat{u}}^i(\cdot),\hat{x}_1^i(\cdot)) + \rho \sum_{i=1}^{\gamma} \mu_i J_2(u^*(\cdot),p_{\hat{u}}^i(\cdot);\hat{x}_2^i(\cdot))$$

$$q_{\hat{u}}^i(t) = y^i(\hat{x}_1^i(t),\hat{u}(t)) \qquad i=1,..,\gamma$$

$$p_{\hat{u}}^i(t) = z^i(\hat{x}_2^i(t),\hat{u}(t)) \qquad i=1,..,\gamma$$

Since $\rho > 0$, $\mu_i > 0$ $i=1,..,\gamma$, either

$$J_1(u^*(\cdot),v_1^{i*}(\cdot);x_1^{i*}(\cdot)) = J_1(\hat{u}(\cdot),q_{\hat{u}}^i(\cdot);\hat{x}_1^i(\cdot)) \qquad i=1,..,\gamma$$

and $J_2(u^*(\cdot),v_2^{i*}(\cdot);x_2^{i*}(\cdot)) = J_2(\hat{u}(\cdot),p_{\hat{u}}^i(\cdot);\hat{x}_2^i(\cdot)) \qquad i=1,..,\gamma \qquad (13)$

or

$$J_1(u^*(\cdot),v_1^{i*}(\cdot);x_1^{i*}(\cdot)) < J_1(\hat{u}(\cdot),q_{\hat{u}}^i(\cdot);x_1^i(\cdot) \text{ for some } i \epsilon \{1,..,\gamma\} \qquad (14)$$

or

$$J_2(u^*(\cdot),v_2^{i*}(\cdot);x_2^{i*}(\cdot)) < J_2(\hat{u}(\cdot),p^i_{\hat{u}}(\cdot);x^i_2(\cdot)) \text{ for some } i\epsilon\{1,..,\gamma\} \tag{15}$$

From (a)

$$\sup_{v\epsilon N(u^*(\cdot))} J_1(u^*(\cdot),v(\cdot);x(\cdot)) = J_1(u^*(\cdot),v_1^{i*}(\cdot);x_1^{i*}(\cdot)) \quad i=1,..,\gamma \tag{16}$$

$$\sup_{v\epsilon N(u^*(\cdot))} J_2(u^*(\cdot),v(\cdot);x(\cdot)) = J_2(u^*(\cdot),v_2^{i*}(\cdot);x_2^{i*}(\cdot)) \quad i=1,..,\gamma \tag{17}$$

Also,

$$J_1(\hat{u}(\cdot),q^i_{\hat{u}}(\cdot);\hat{x}^i_1(\cdot)) \le \sup_{v(\cdot)\epsilon N(\hat{u}(\cdot))} J_1(\hat{u}(\cdot),v(\cdot);x(\cdot)) \tag{18}$$

$$J_2(\hat{u}(\cdot),p^i_{\hat{u}}(\cdot);\hat{x}^i_2(\cdot)) \le \sup_{v(\cdot)\epsilon N(u(\cdot))} J_2(\hat{u}(\cdot),v(\cdot),x(\cdot)) \tag{19}$$

Using (16)-(19), each of the three cases (13), (14), and (15) imply $u^*(\cdot)$ is Pareto optimal.

5. DISCUSSION

For a given initial state and time, one can proceed as follows to obtain Pareto optimal solutions. Form the two-player zero-sum game (4)-(6). For those values of α_1 and α_2 with $\alpha_1 > 0$, $\alpha_2 > 0$ and $\alpha_1 + \alpha_2 = 1$ for which there is an open loop saddle point, the corresponding $u(\cdot)$ is, according to Theorem 3.1, a Pareto optimal solution.

If there does not exist a saddle point solution, we must resort to Theorem 4.1. Choose a real integer $\gamma \ge 2$ and functions $y^i(\cdot,\cdot)$, $z^i(\cdot,\cdot)$, $i=1,..,\gamma$. Then determine the optimal control for the problem $P(\gamma,\rho,\mu_i,y^i(\cdot,\cdot),z^i(\cdot,\cdot))$ in terms of ρ and μ_i, i.e. $u(\cdot,\rho,\mu_1,\ldots,\mu_\gamma)$. If possible, choose the μ_i so that

$$J_1(u(\cdot),y^i(\cdot,\cdot);x^i_1(\cdot)) + \rho\, J_2(u(\cdot),z^i(\cdot,\cdot);x^i_2(\cdot))$$
$$= J_1(u(\cdot),y^j(\cdot,\cdot);x^j_1(\cdot)) + \rho\, J_2(u(\cdot),z^j(\cdot,\cdot);x^j_2(\cdot))$$

for all i, $j\epsilon\{1,..,\gamma\}$. For every $\rho > 0$ for which this can be done, we then have a candidate for a Pareto optimal solution of the problem (1)-(3). It may then be possible to verify that these candidates are solutions by using Theorem 4.1. In checking (a) and (b), of Theorem 4.1, standard sufficiency results for optimal control, such as [10-14], can be used. Of course, choosing γ, $y^i(\cdot,\cdot)$ and $z^i(\cdot,\cdot)$ may be difficult and some ingenuity as well as trial and error must be used.

The general idea of the procedure is to let α_1 take on all possible values in the interval (0,1). For every value of α_1 in this interval we hope to get a Pareto optimal solution. If there is an open loop saddle point solution corresponding to a particular α_1, we use Theorem 3.1; if not, we find a minmax solution and use Theorem 4.1.

6. EXAMPLE

Consider the following problem with scalar $x(\cdot)$, $u(\cdot)$ and $v(\cdot)$

$$J_1(u(\cdot),v(\cdot);x(\cdot)) = x^2(1),\ J_2(u(\cdot),v(\cdot);x(\cdot)) = -x(1) \tag{20}$$

$$\dot{x}(t) = u(t)v(t), \qquad x(0) = \frac{1}{2} \tag{21}$$

$$U = \{u : u^2 \leq 1\}\ , \qquad V = \{v : 1 \leq v \leq 2\} \tag{22}$$

First we form the two player differential game (4)-(6)

$$\hat{J}(u(\cdot),v_1(\cdot),v_2(\cdot);x_1(\cdot),x_2(\cdot)) = \alpha_1 x_1^2(1) - \alpha_2 x_2(1)$$

$$\dot{x}_1(t) = u(t)v_1(t)\ ; \qquad x_1(0) = \frac{1}{2}$$

$$\dot{x}_2(t) = u(t)v_2(t)\ ; \qquad x_2(0) = \frac{1}{2}$$

Checking for open loop saddle solutions as α_1 ranges between 0 and 1, we find that there are such solutions for $\alpha_1 \epsilon (0,\ 6/7)$ and that the controls $u(\cdot)$, $v_1(\cdot)$ and $v_2(\cdot)$ are constant. These results are summarized in Table 1.

TABLE 1

α	$\hat{u}(\cdot)$	$\hat{v}_1(\cdot)$	$\hat{v}_2(\cdot)$
$0 < \alpha_1 \leq \frac{1}{11}$	1	2	1
$\frac{1}{11} < \alpha_1 \leq \frac{1}{3}$	$\frac{1 - 2\alpha_1}{8\alpha_1}$	2	1
$\frac{1}{3} < \alpha_1 \leq \frac{1}{2}$	0	$\frac{1 - \alpha_1}{\alpha_1}$	1
$\frac{1}{2} < \alpha_1 \leq \frac{2}{3}$	0	$\frac{2(1 - \alpha_1)}{\alpha_1}$	2
$\frac{2}{3} < \alpha_1 \leq \frac{6}{7}$	$\frac{1 - \frac{3}{2}\alpha_1}{\alpha_1}$	1	2

The inequality $\hat{J}(\hat{u}(\cdot),\hat{v}_1(\cdot),\hat{v}_2(\cdot)) \leq \hat{J}(u(\cdot),\hat{v}_1(\cdot),\hat{v}_2(\cdot))$was verified via the direct sufficient conditions of [13,14] while the inequality $\hat{J}(\hat{u}(\cdot),\hat{v}_1(\cdot),\hat{v}_2(\cdot)) \geq \hat{J}(\hat{u}(\cdot),\hat{v}_1(\cdot),v_2(\cdot))$ was established using field type sufficiency theorems [10,11]. That the $\hat{u}(\cdot)$ part of the saddle point solution is a Pareto optimal solution follows from Theorem 3.1.

For $6/7 < \alpha_1 < 1$, there is no open loop saddle point solution and we must resort to Theorem 4.2. To this end, we consider the control problem given by (11)-(12) with $\gamma=2$, $y^1(\cdot) = 1$, $y^2(\cdot) = 1$, $z^1(\cdot) = 2$, $z^2(\cdot) = 2$, $0 < \rho < \frac{1}{6}$, $\mu_1 = 2\rho + \frac{2}{3}$ and $\mu_2 = 1-\mu_1$.

$$K(u(\cdot),y^i(\cdot),z^i(\cdot);x_1^i(\cdot),x_2^i(\cdot)) = \mu_1[x_1^1(1)]^2 + \mu_2[x_1^2(1)]^2 - \rho\mu_1 x_2^1(1) - \rho\mu_2 x_2^2(1)$$

$$\dot{x}_1^1(t) = u(t) \qquad , \qquad \dot{x}_2^1(t) = 2u(t)$$

$$\dot{x}_1^2(t) = 2u(t) \qquad , \qquad \dot{x}_2^2(t) = 2u(t)$$

$$x_1^1(0) = x_1^2(0) = x_2^1(0) = x_2^2(0) = \frac{1}{2}$$

Since $u^*(t) = -1/3$ is a solution to this optimal control problem and since $y^i(\cdot) \epsilon\ \mathcal{L}_1(u^*(\cdot))$, $z^i(\cdot) \epsilon\ \mathcal{L}_2(u^*(\cdot))$, $i=1,2$, $u^*(\cdot)$ is a Pareto optimal solution for this problem. Also, since $\rho = \alpha_2/\alpha_1$ and $0 < \rho < 1/6$, this solution corresponds to $6/7 < \alpha_1 < 1$. Note that Theorem 3.1 does not apply when $6/7 < \alpha_1 < 1$ since there is no saddle point solution and Theorem 4.1 is needed.

REFERENCES

1. Goffin, J. and Haurie, A., "Necessary Conditions and Sufficient Conditions for Pareto Optimality in a Multicriterion Perturbed System," Proceedings of the 5th IFIP Conference on Optimization, 1973.
2. Goffin, J. "Conditions for Pareto Optimality in a Multicriterion System with Nondifferentiable Functions," Proceedings of the 11th Annual Allerton Conference on Circuits and System Theory, 1973.
3. Schmitendorf, W. E. "Static Multicriteria Problems: Necessary Conditions and Sufficient Conditions," Proceedings IFAC Symposium on Large Scale Systems, Udine, Italy, June 16-20, 1976.
4. Schmitendorf, W. E. and Moriarty, G. "A Sufficiency Condition for Coalitive Pareto-Optimal Solutions," J. of Opt. Theory and Applications, Vol. 18, No. 1, 1976.
5. Bryson, A. E. and Ho, Y. C. Applied Optimal Control, Blaisdell, Waltham, Mass. 1969.
6. Schmitendorf, W. E. "Existence of Optimal Open-Loop Strategies for a Class of Differential Games," J. of Opt. Theory and Applications, Vol. 5, No. 5, 1970.
7. Schmitendorf, W. E. "Differential Games with Open-Loop Saddle Point Conditions," IEEE Trans. Auto. Control, Vol. AC-15, No. 3, 1970.
8. Schmitendorf, W. E. "A Sufficient Condition for Minmax Control of Systems with Uncertainty in the State Equations," IEEE Trans. Auto. Control, August, 1976.
9. Schmitendorf, W. E. "A General Sufficiency Condition for Minmax Control," (to be submitted for publication).
10. Leitmann, G. "Sufficiency Theorems for Optimal Control," J.Opt. Theory and Application, Vol. 2, pp. 285-292, 1968.
11. Leitmann, G. "A Note on a Sufficiency Theorem for Optimal Control," J. Opt. Theory and Application, Vol. 3, pp. 76-78, 1969.
12. Stalford, H. "Sufficient Conditions for Optimal Control with State and Control Constraints," J. Opt. Theory and Application, Vol. 7, pp. 118-135, 1971.
13. Mangasarian, O. L. "Sufficient Conditions for the Optimal Control of Nonlinear Systems," SIAM J. on Control, Vol. 4, No. 1, 1966.
14. Leitmann, G., and Stalford, H. "A Sufficiency Theorem for Optimal Control," J. of Opt. Theory and Applications, Vol. 8, No. 3, 1971.

EQUILIBRIUM CONCEPTS IN STOCHASTIC MULTICRITERIA OPTIMIZATION

DAVID A. CASTANON
Electronic Systems Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

This paper explores some issues in equilibrium solutions of stochastic multicriteria decision problems. Correct treatment of this problem involves an extended notion of state, thereby altering conventional concepts of open-loop and closed-loop decisions. Restrictions on transfer of information lead to signalling-free concepts. These topics are discussed and illustrated with a simple example.

1. INTRODUCTION

A common approach in the study of decentralized decision-making in optimization problems is to model the individual decision makers as players within a game [2]. Such an approach is appropriate for classes of systems where there are multiple criteria for multiple decision makers, decentralized information, and natural hierarchies in the order of decision-making. Recent works in the field [1], [5], [6], have focused on systems with a sequential decision structure; these decisions are commonly referred to as "Stackelberg" solutions. Most of these works have dealt with deterministic systems, or with systems where the stochastic structure is very simple. This paper discusses solutions of systems with sequential decision structure and nontrivial stochastic information structure. In particular, it defines and examines three classes of equilibrium solutions, providing a simple example which highlights their differences.

2. DEFINITIONS

This section formally defines some basic terms, and formulates the sequential decision problem.

Definition 2.1. An n-player game is a triple G: $\{\underline{D}, J, SR\}$, where $\underline{D} = D_1 \times D_2 \times \ldots \times D_N$ is a decision space, D_i is the set of decisions for player i, J is a vector, extended-real valued performance criteria.

$$J: D \rightarrow (R^e)^N$$

J_i is the criterion of the i^{th} player. SR is a selection rule which selects a subset $S^* \subset \underline{D}$ as a solution of the game.

The essential problem of solving a game consists of obtaining S*. The selection rule (also known as solution concept) is usually based on behavioral principles governing the players in the game; examples of these principles are Nash, Pareto, minimax, and Stackelberg principles.

Definition 2.2. An Equilibrium selection rule is a selection rule such that

$$\underline{d}^* = (d_1^*, \ldots d_N^*) \in S^*$$

if $J_i(\underline{d}^*) \leq J_i(\underline{d}_i)$ for all $\underline{d}_i \in \underline{D}$ of the form

$$\underline{d}_i = (d_1^*, \ldots, d_{i-1}^*, d_i, d_{i+1}^*, \ldots d_N^*) \quad \text{for all } i = 1, \ldots N.$$

The preparation of this paper was supported in part by ONR Contract N00014-76-C-0346.

Equilibrium solutions are thus defined in terms of performance criteria and admissible strategies. Consider now the following stochastic sequential dynamical system:

(A) State equation

$$x_{t+1} = f_t(x_t, u_t^1, \dots, u_t^n, \theta_t) \tag{2.1}$$

Observation equation

$$y_t^i = g_t^i(x_t, \xi_t^i) \tag{2.2}$$

Loss functionals

$$L_0^i = \sum_{t=0}^{T-1} h_t^i(x_t, u_t^1, \dots, u_t^n) + h_T^i(x_T) \tag{2.3}$$

The variables x_t, y_t^i, u_t^i are elements of appropriately defined state, observation and decision spaces, assumed to be finite-dimensional Euclidean spaces. The variables θ_t, ξ_t^i and x_0 are random variables, with the functions f_t, g_t^i and h_t^i assumed appropriately measurable. The active information available to each decision-maker i at time t is denoted I_t^i, where

$$I_0^i = \{y_0^i, u_0^1, \dots u_0^{i-1}\} \tag{2.4}$$

$$I_t^i = I_{t-1}^i \cup \{y_t^i, u_{t-1}^{i+1}, \dots u_{t-1}^n, u_t^0, \dots u_t^{i-1}\} \tag{2.5}$$

Each player knows the value of the previous decisions made by all players. At each stage, decision are made and announced sequentially. This sequential information pattern is typical of Stackelberg dynamic games [3], [4]. The stochastic sequential dynamical system described in equations (2.1) to (2.5) is a typical model of hierarchical system [6]. However, this mathematical description does not constitute a game; performance criteria, admissible strategies and selection rules must be specified.

Definition 2.3. An open-loop strategy λ_t^i is a measurable map from the space of information sets to the space of decision such that

$$u_t^i = \lambda_t^i(I_t^i).$$

Definition 2.4. An open-loop equilibrium solution to the dynamic decision model (A) is a member of the equilibrium solution set of the game G_{OL}, where

$$G_{OL} := \{\underline{D}_{OL}, \underline{J}, SR_{eq}\}$$

and

$$D_{OL}^i = \{\text{open-loop } \underline{\lambda}^i = (\lambda_0^i, \dots \lambda_{T-1}^i)\}$$

$$\underline{\lambda} = (\underline{\lambda}^1, \dots \underline{\lambda}^n)$$

$$J^i(\underline{\lambda}) = E^{\underline{\lambda}}(L^i)$$

and $E^{\underline{\lambda}}$ is the expectation with respect to the probability distribution induced by the strategy $\underline{\lambda}$.

Notice that, even through the admissible strategies are functions of available information, the solution is termed open-loop. This distinction is made because the information sets I_t^i do not truly represent the state of the system; they do not include knowledge of the previous strategies used in the game by other players.

Definition 2.5: A closed-loop strategy γ^i_t is a strategy-dependent map which maps $\{\underline{\gamma}^i_t\}$ and I^i_t into an admissible decision u^i_t, where

$$\underline{\gamma}^i_t = \{\gamma^1_0, \ldots \gamma^n_0, \gamma^1_1, \ldots \gamma^n_{t-1}, \gamma^1_t, \ldots \gamma^{i-1}_t\}$$

$$u^i_t = \gamma^i_t(I^i_t; \underline{\gamma}^i_t)$$

The key difference between open- and closed-loop strategies is the dependence of closed-loop strategies on previous strategies, because the true state of the system is a probability distribution (conditional or unconditional) depending on previous strategies.

Definition 2.6: A closed-loop equilibrium solution to the dynamic decision model (A) is a member of the equilibrium solution set of the game G_{CL}, where

$$G_{CL}: = \{\underline{D}_{CL}, \underline{J}_{CL}, SR_{eq}\}$$

and $D^i_{OL}: = \{\text{closed-loop } \underline{\gamma}^i = (\gamma^i_0, \ldots \gamma^i_{T-1})\}$

$$\underline{\gamma} = (\underline{\gamma}^1, \ldots \underline{\gamma}^n)$$

$$J^i(\underline{\gamma}) = E^{\underline{\gamma}}(L^i)$$

$E^{\underline{\gamma}}$ is the expectation with regard to the induced probability distribution.

Occasionally, the dependence of closed-loop strategies on previous strategies may be summarized through a transformation to a standard form, as in Witsenhausen [7], or Castanon and Sandell [1]. The original stochastic model is transformed to an equivalent deterministic model, defining the unconditional probability distribution of the state, x_t, denoted by π_t, as the true state.

The closed-loop strategies are represented as

$$u^i_t = \gamma^i_t(I^i_t; \pi_t)$$

The equilibrium solutions obtained with this representation of admissible strategies are included in the equilibrium solutions obtained in definition (2.6); the converse is not true in general.

This approach requires a non-trivial transformation of the system dynamics to a standard form, which is not always possible to do. An alternative approach is to use another representation of closed-loop strategies, making γ^i_t dependent on the conditional distribution of the state, given information I^i_t and previous strategies $\underline{\gamma}^i_t$. That is,

$$u^i_t = \gamma^i_t(\hat{\pi}^i_t, I^i_t)$$

where

$$\hat{\pi}^i_t = \text{probability distribution of } x_t\text{, given } I^i_t \text{ and } \underline{\gamma}^i_t$$

The form of closed-loop strategies implies that the players have exact knowledge of all strategies used by other players previously. This enables the players to extract information about the system state from observation of the previous decisions. This is known as signalling [4]. In many situations, it is impractical or inaccurate to use strategies which require exact knowledge of past strategies; furthermore, it may be desirable for players to choose their strategies on the basis of internal perceptions of cost (delayed commitment) rather than an external cost such as the unconditional expectation of the loss function.

Definition 2.7: A signalling-free conditional probability distribution for player i at stage t, denoted p_t^i, is a version of the conditional probability distribution of the system variables assuming all $u_k^j \varepsilon I_t^i$ were produced by constant strategies.

Definition 2.8: A signalling-free equilibrium solution to the dynamic decision model (A) is a member of the equilibrium solution set of the game G_{SF}, where

$$G_{SF} : = \{\underline{D}_{SF}, \underline{J}_{SF}, SR_{eq}\}$$

$$D_{SF}^i : = \{\gamma_t^i(I_t^i; p_t^i)\}$$

$$\underline{\gamma}_t^i = \{\gamma_t^i, \ldots \gamma_t^n, \gamma_{t+1}^1, \ldots \gamma_T^n\}$$

$$J_t^i(\underline{\gamma}_t^i) = E^{p_t^i, \underline{\gamma}_t^i}(L^i)$$

where $E^{p_t^i, \underline{\gamma}_t^i}$ is the expectation with respect to the conditional probability distribution p_t^i, assuming future decisions are defined using strategies $\underline{\gamma}_t^i$.

A key point to note is that signalling-free equilibrium solutions are equilibria between nT players, reflecting the fact that decisions are made stagewise based on internal perceptions. Also, the equilibrium properties of the strategies are not lost if at any prior stage, non-equilibrium strategegeis were used. This is possible because of the lack of signalling; the solution (and the relevant probability densities) can be obtained recursively dealing exclusively with the information sets I_t^i and future strategies $\underline{\gamma}_t^i$.

3. ILLUSTRATIVE EXAMPLES

Consider the discrete-time system described by the state equations

$$x_1 = x_0 + u_0 + v_0 + \theta_0 \tag{3.1}$$

$$x_2 = x_1 + u_1 + \theta_1 \tag{3.2}$$

with observations

$$z_0 = x_0 + \xi_0 \tag{3.3}$$

$$z_1 = x_1 \tag{3.4}$$

where x_0, θ_0, θ_1, ξ_0 are independent zero-mean Gaussian random variables with covariances Σ_0, Θ_0, Θ_1, Ξ_0 respectively, and all variables are real-valued scalars.

The players' active information sets are

$$I_0^1 = \{z_0\} \tag{3.5}$$

$$I_1^1 = \{x_1, z_0, v_0\} \tag{3.6}$$

$$I_0^2 = \{u_0\} \tag{3.7}$$

where player u is denoted 1, player v is denoted 2. The loss functionals are

$$L^1 = x_2^2 + u_0^2 + u_1^2 \tag{3.8}$$

$$L^2 = x_2^2 + v_0^2 \tag{3.9}$$

The linear quadratic Gaussian nature of this example implies that there are open-loop and closed-loop stochastic equilibria of the form

$$u_0 = az_0 \tag{3.10}$$

$$v_0 = bu_0 \tag{3.11}$$

$$u_1 = cx_1 \tag{3.12}$$

Additionally, the signalling-free equilibria will also be in that form. Appendix A computes the signalling-free equilibrium strategies and the associated unconditional losses. Appendices B and C compute the linear closed-loop and open-loop strategy and the unconditional losses, respectively. The results are summarized in Table 3.1 when $\Xi_0 = \Sigma_0 = \Theta_0 = \Theta_1 = 1$.

Table 3.1

	Signalling-free Equilibrium	Closed-loop Equilibrium	Open-loop Equilibrium
a*	$-\frac{5}{33}$	$-\frac{4}{33}$	$-\frac{2}{33}(\frac{3}{2} \pm\sqrt{3})$
b*	$-\frac{1}{5}$	$\frac{5}{8}$	$\frac{-1 \pm\sqrt{3}}{2}$
c*	$-\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$
J^1	1.94	1.87	1.86 or 2.03
J^2	1.45	1.43	1.46 or 1.51

4. CONCLUSION

The concepts of open-loop and closed-loop strategies in sequential stochastic games were adapted from deterministic games with one principle in mind: Closed-loop strategies are maps which depend on the state of the system. In stochastic systems, this state is often a probability distribution. These concepts also apply to non-sequential games. The example in Section 3 and Appendices B and C highlight the differences between open-loop and closed-loop equilibrium strategies. In particular, open-loop equilibria are obtained using variational principles; closed-loop equilibria can be obtained through stagewise decomposition similar to dynamic programming.

A third class of equilibrium solutions was defined, based on the concept of "delayed commitment" and no signalling. This solution is easier to compute in general because of the restrictions on signalling and the recursive nature of the definition.

There are many open areas of research. Castanon [4] has established an equivalence between signalling-free and closed-loop equilibria for special information patterns. It is not clear that open-loop equilibria are also closed-loop equilibria in stochastic games, as is true in deterministic games [8]. The dependence of closed-loop equilibria on the representation of the true system state must be studied in depth. Understanding these questions is essential in formulating a unified theory of generalized decision-making.

APPENDIX A

Consider the problem of obtaining a signalling-free equilibrium for the system described in Section 3.

$$J_1^1 = E\{u_1^2 + x_2^2 | x_1\} = u_1^2 + (x_1 + u_1)^2 + \Theta_1 \tag{A.1}$$

Hence,

$$u_1^* = -\frac{1}{2} x_1 \tag{A.2}$$

Similarly,

$$\begin{aligned} J_0^2 &= E\{v_0^2 + x_2^2 | u_0 = \text{constant},\ u_1 = -\frac{1}{2} x_1\} \\ &= \Theta_0 + \Theta_1 + \frac{1}{4}(u_0 + v_0)^2 + v_0^2 + \Sigma_0 \end{aligned} \tag{A.3}$$

because x_0 is zero-mean. Thus,

$$v_0^* = -\frac{1}{5} u_0 \tag{A.4}$$

For player u at stage 0, one gets

$$\begin{aligned} J_0^1 &= E\{x_2^2 + u_1^2 + u_0^2 | z_0,\ v_0 = -\frac{1}{5} u_0,\ u_1 = -\frac{1}{2} x_1\} \\ &= \frac{1}{2} \bar{x}_1^2 + u_0^2 + \Sigma_1^0 \end{aligned} \tag{A.5}$$

where

$$\hat{x}_0 = \frac{\Sigma_0}{\Sigma_0 + \Xi_0} z_0 \tag{A.6}$$

$$\bar{x}_1 = \hat{x}_0 + \frac{4u_0}{5} \tag{A.7}$$

$$\Sigma_1^0 = \frac{\Sigma_0 \Xi_0}{\Sigma_0 + \Xi_0} + \Theta_0 \tag{A.8}$$

Hence,

$$u_0^* = \frac{-10}{33} \hat{x}_0 = \frac{-10}{33} \frac{\Sigma_0}{\Sigma_0 + \Xi_0} z_0 \tag{A.9}$$

Assuming $\Sigma_0 = \Xi_0 = \Theta_0 = \Theta_1 = 1$, the unconditional costs can be computed as:

$$E\{u_0^2\} = \frac{1}{2} \cdot \frac{10}{33}^2 \tag{A.10}$$

$$E\{v_0^2\} = \frac{1}{2} \cdot \frac{2}{33}^2 \tag{A.11}$$

$$E\{x_1^2\} = 2 - \frac{8 \cdot 29}{(33)^2} \tag{A.12}$$

Thus,

$$J_1 = 2 + \frac{1}{2}\left(\frac{10}{33}\right)^2 - \frac{4 \cdot 29}{(33)^2} = 2 - \frac{2}{33} \tag{A.13}$$

$$J_2 = \frac{3}{2} - \frac{58}{(33)^2} + \frac{2}{(33)^2} = \frac{3}{2} - \frac{56}{(33)^2} \tag{A.14}$$

APPENDIX B

Consider the problem of obtaining a closed-loop equilibrium solution for the model in Section 3, using strategies with conditional probability distribution dependence.

$$J_1^1 = E\{u_1^2+x_2^2 | a,\ b,\ z_0,\ x_1\} = u_1^2 + (x_1+u_1)^2 + \Theta_1 \tag{B.1}$$

Thus,

$$u_1^* = -\frac{1}{2}x_1 \tag{B.2}$$

$$J_0^2 = E\{x_2^2 + u_0^2 | az_0 = u_0,\ u_1 = -\frac{1}{2}x_1\} \tag{B.3}$$

If a=0, then b=0. Thus, assume a ≠ 0. In this case, knowledge of u_0 implies knowledge of z_0. Thus

$$J_0^2 = \frac{\bar{x}_1^2}{4} + v_0^2 + \Theta_1 + \Sigma_1^0 \tag{B.4}$$

where

$$\hat{x}_0 = \frac{\Sigma_0}{\Sigma_0+\Xi_0}\ z_0 \tag{B.5}$$

$$\bar{x}_1 = \hat{x}_0 + u_0 + bu_0 \tag{B.6}$$

$$\Sigma_1^0 = \frac{\Sigma_0\Xi_0}{\Sigma_0+\Xi_0} + \Theta_0 \tag{B.7}$$

Thus,

$$v_0^* = -\frac{1}{5}(\hat{x}_0 + az_0) = -\frac{1}{5}(az_0)(1 + \frac{\Sigma_0}{a(\Sigma_0+\Xi_0)}) \tag{B.8}$$

Now,

$$J_0^1 = E\{x_2^2+u_0^2+u_1^2 | z_0,\ u_1 = -\frac{1}{2}x_1,\ v_0 = -\frac{1}{5}u_0(b^*)\}$$

$$= \frac{\bar{x}_1^2}{2} + u_0^2 + \Sigma_1^0 \tag{B.9}$$

where

$$\bar{x}_1 = \frac{4}{5}(u_0 + \hat{x}_0) \tag{B.10}$$

so,

$$u_0^* = -\frac{8}{33}z_0\frac{\Sigma_0}{\Sigma_0+\Xi_0} \tag{B.11}$$

When $\Xi_0 = \Theta_0 = \Theta_1 = \Sigma_0 = 1$, then, the unconditional cost is given by

$$E\{u_0^2\} = (\frac{8}{33})^2 \cdot \frac{1}{2} \tag{B.12}$$

$$E\{v_0^2\} = (\frac{5}{33})^2 \cdot \frac{1}{2}) \tag{B.13}$$

$$E\{x_1^2\} = 2(1 - \frac{53\cdot 13}{(66)^2} \tag{B.14}$$

$$J_1 = 2 - \frac{187}{(22)(66)} \tag{B.15}$$

$$J_2 = \frac{3}{2} - \frac{589}{2(66)^2} \tag{B.16}$$

APPENDIX C

Consider the problem of obtaining a closed-loop equilibrium solution for the model in Section 3, using strategies with unconditional probability distribution dependence. The equivalent deterministic problem is

$$\Sigma_1 = \Sigma_0(1 + a + ab)^2 + \Xi_0(1+b)^2 a^2 + \Theta_0 \tag{C.1}$$

$$\Sigma_2 = \Sigma_1(1+c)^2 + \Theta_1 \tag{C.2}$$

$$J_1 = \Sigma_2 + c^2\Sigma_1 + a^2(\Sigma_0 + \Xi_0) \tag{C.3}$$

$$J_2 = \Sigma_2 + a^2b^2(\Sigma_0 + \Xi_0) \tag{C.4}$$

Using dynamic programming for the model of equations (C.1)-(C.4) yields

$$c^* = -\frac{1}{2} \tag{C.5}$$

$$b^* = -\frac{1}{5}\left(1 + \frac{\Sigma_0}{a(\Sigma_0+\Xi_0)}\right) \tag{C.6}$$

$$a^* = -\frac{8}{33}\ \frac{\Sigma_0}{\Sigma_0+\Xi_0} \tag{C.7}$$

This solution is exactly the same solution obtained in Appendix B.

Now consider the problem of finding an open-loop equilibrium. Using the equivalent deterministic model of equations (C.1) to (C.4), one obtains necessary conditions

$$c^* = \frac{1}{2} \tag{C.8}$$

$$\frac{\partial J_1}{\partial a} = 0 \tag{C.9}$$

$$\frac{\partial J_2}{\partial b} = 0 \tag{C.10}$$

$$\frac{\partial J_1}{\partial a} = 2(\Sigma_0+\Xi_0)a + (\Sigma_0(1+b)(1+a+ab) + \Xi_0(1+a)(a+ab) \tag{C.11}$$

$$\frac{\partial J_2}{\partial b} = a\,\frac{\Sigma_0}{2}\,(1+a+ab) + \frac{\Xi_0}{2}\,a(a+ab) + 2a^2b(\Sigma_0+\Xi_0) \tag{C.12}$$

Let $g = \dfrac{\Sigma_0}{\Sigma_0+\Xi_0}$. Then, (C.8)-(C.12) imply

$$2a + (1+b)^2a + g(1+b) = 0 \tag{C.13}$$

$$4a^2b + a^2(1+b) + ga = 0 \tag{C.14}$$

Assuming $a \neq 0$, then

$$b = -\frac{1}{2} \pm\frac{\sqrt{3}}{2} \tag{C.15}$$

$$a = -\frac{4g}{33}\left(\frac{3}{2} \pm\sqrt{3}\right) \tag{C.16}$$

The solution costs, assuming $\Theta_0 = \Theta_1 = \Sigma_0 = \Xi_0 = 1$, are

$$J_1 = 2 - \left(\frac{1}{11}\right)^2\left(\frac{27}{4} \pm 6\sqrt{3}\right) \tag{C.17}$$

$$J_2 = \frac{3}{2} - \frac{1}{2}\left(\frac{1}{11}\right)^2\left(\frac{49}{12} \pm \frac{10\sqrt{3}}{3}\right) \tag{C.18}$$

REFERENCES

1. D. Castanon and N.R.Sandell,Jr.,"Equilibrium Solutions in Sequential Stochastic Games", IEEE Trans. on Auto. Control, Vol. AC-22, No. 4, August, 1977.
2. Y.C.Ho,"on the Minimax Principle and Zero-sum Stochastic Differential Games," JOTA, Vol. 13, No. 3, March 1974.
3. M. Simaan and J.B.Cruz Jr., "Additional Aspects of the Stackelberg Strategy in Nonzero-Sum Games," JOTA, Vol. 11, no.6,June 1973.
4. D. Castañon, "Equilibria in Stochastic Dynamic Games of Stackelberg Type," Report ESL-R-662, Electronics Systems Laboratory,MIT,May 1976.
5. S. Glankwamdee and J.B.Cruz, Jr., "Decentralized Stackelberg Strategies for Interconnected Stochastic Dynamic Systems," Report DC-1, Decision and Control Laboratory, CSL and EE,University of Illinois,April 1977.
6. J. B. Cruz, Jr., "Stackelberg Strategies for Multilevel Systems," in Directions in Large Scale Systems, Y. C. Ho and S. K. Mitter, editors, Plenum Publishing Corp., New York, 1976.
7. H. S. Witsenhausen, "A Standard Form for Sequential Stochastic Control," Mathematical Systems Theory, Vol. 7, 1973.
8. N. Sandell Jr., "On Open Loop and Closed Loop Nash Strategies," IEEE Trans. on Auto. Control, Vol. AC-19, June 1974.
9. M. Simaan and J.B.Cruz,Jr., "On the Stackelberg Strategy in Nonzero-Sum Games," JOTA, vol. 11, no. 5, 1973.

MACRO-ECONOMIC STABILIZATION POLICY FOR AN UNCERTAIN DYNAMIC ECONOMY

G. LEITMANN
University of California
Berkeley, California

H. Y. WAN, Jr.
Cornell University
Ithaca, New York

ABSTRACT

A stabilization policy is developed for an economy with unknown and possibly unknowable characteristics. Uniform asymptotic stability is guaranteed for all initial states and for all admissible uncertainties. Welfare loss over the infinite horizon is bounded by a level dependent on the initial state. The policy is robust, with a simple structure. A simple example sheds light on the fiscalist-monetarist debate.

I. INTRODUCTION

In this paper we discuss the stabilization of an economic system with some unknown and possibly even unknowable characteristics. We adopt a simple two variable model that serves to illustrate the method proposed and also to shed some light on certain policy debates in macro-economic theory. In contrast to the technique employed here, the sizable and growing literature on macro-economic stabilization, with few exceptions, is based on the assumption that statistical information (e.g., mean, variance) is known concerning the distributions of system parameters and noise, either by observation or estimation through adaptive methods. Moreover, most of these studies lead to quantitative results only. Hence, they do not address issues raised by Friedman and his critics or the controversy between "monetarists" and "fiscalists" in a general, qualitative manner. An excellent survey of the literature may be found in Kendrick (1976).

Before presenting our model, some discussion may be indicated regarding the two main points which distinguish it from most of the others considered elsewhere.

1. The "knowability" of system characteristics. An economic system encompasses the activities of many firms and individuals whose actions are conditioned by their anticipations of how government will act. Some advocates of stochastic optimization propose a successive approximation procedure. The government should act in accordance with "prior" information about the distributions of system parameters and noise (disturbances). In an economic system in which uncertainty prevails in the sense of Heisenberg, the invoking of adaptive (learning) procedures may be conceptually unsuitable (which "posterior" distributions are the true ones?) and practically inappropriate (there may be no expectational equilibrium, or successive approximations may result in an equilibrium that is far from "best" in terms of some governmental payoff or performance index).

On the other hand, here we consider some system parameters to be unknown functions of time and of the state of the economy. Provided these parameters take on values in bounded intervals and the bounds are sufficiently small relative to control capabilities, under rather mild conditions one can guarantee asymptotically stable behavior of the system and a well defined bound for the "loss function" over an infinite horizon. Furthermore, the policy functions are given in closed "feedback" form, that is, as functions of the state of the economy. The method employed here is based on that of Gutman and Leitmann (1975, 1976) for stabilization of uncertain dynamic systems; details are given in Leitmann and Wan (1977).

The only other treatment, known to us, that addresses economic stabilization from the point of view of bounded uncertainty is the one by Chang and Stekler (1977); they approach the problem via fuzzy dynamic programming. However, theirs is a discrete, finite horizon model and so they do not provide for guaranteed stability. Furthermore, their policy prescription is not in closed form and consequently cannot be employed to gain qualitative insights into macro-economic debates.

2. Qualitative insights. Friedman (1956, 1959) points out that monetary control involves an uncertain "lag in effect". When applied against inflation or recession, it may be ineffective in combatting the current problem because of the delay in its effect, or its effect may come long after it is needed and only exacerbate the current problem. The model proposed by us here provides a simple mechanism for evaluating the following questions:

(a) In advocating abstention from employing monetary policy against cyclical economic disturbances, is Friedman always right, never right, or right under certain conditions? If the last possibility occurs, what are the conditions vindicating Friedman?

(b) Without assessing empirical evidence, it is important to determine the effect on the utility of the anti-cyclical policy instruments of their characteristics (uncertainty in the fiscal "policy multiplier" versus uncertainty in the monetary "lag in effect"). What is this effect?

We postulate a very simple model in the hope of arriving at easily comprehensible insights. At least in principle, there appears to be no difficulty in enlarging the model by incorporating more state variables or more control instruments.

II. THE MODEL

Let y_t and $\tilde{y}_t$, respectively, denote the actual and target "aggregate demand" levels at time t, evaluated at some constant price. Suppose that inflation level, I_t , and unemployment level U_t , depend on y_t and t :

$$I_t(y_t) \triangleq I(y_t, t) \tag{1}$$

$$U_t(y_t) \triangleq U(y_t, t) \tag{2}$$

and consider an "instantaneous loss" function, $G_o(\cdot)$, of I_t , U_t and t :

$$\Gamma_t(y_t) \triangleq \Gamma(y_t, t) \triangleq G_o(I_t, U_t, t) \tag{3}$$

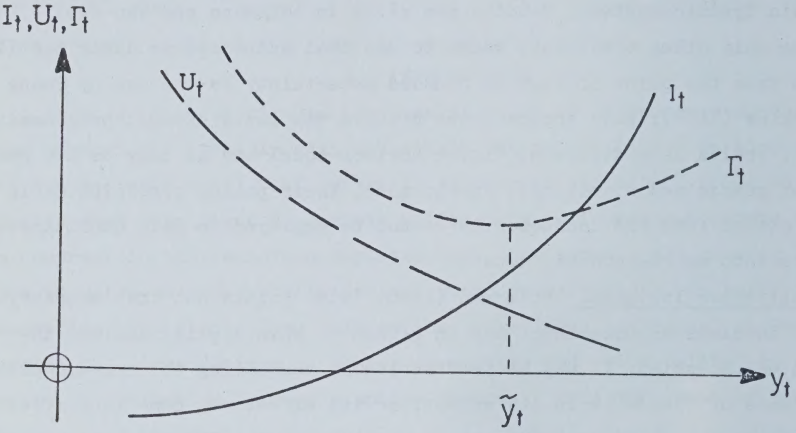


Figure 1. Parameterized Phillips Curve

Plotting I_t as a function of U_t results in the "Phillips curve" which depicts the alternatives open to the planner at time t . Figure 1 depicts this trade-off parametrically with respect to aggregate demand, y_t . The shift in the Phillips curve is accomplished by means of the time variable, t , in $I(\cdot)$ and $U(\cdot)$. The target demand level at time t , $\tilde{y}_t$, is then defined as the one minimizing the instantaneous loss $\Gamma(y_t, t)$, as shown schematically in Figure 1. This formulation allows for the co-existence of inflation and unemployment. Now we introduce an "avoidable loss" that depends on $\Gamma(y_t, t)$ and $\Gamma(\tilde{y}_t, t)$, and hence on y_t, $\tilde{y}_t$ and t. Generally one may wish to let the avoidable loss be $\Gamma(y_t, t) - \Gamma(\tilde{y}_t, t)$. In particular, we shall suppose that the "avoidable loss" function, $G(\cdot)$, is a non-negative, continuous function of $y_t - \tilde{y}_t$ and t , for which there exist positive constants α , β and T such that

$$G(y_t - \tilde{y}_t, t) \leqslant \alpha(y_t - \tilde{y}_t)^2 \quad \text{for} \quad |y_t - \tilde{y}_t| \leqslant \beta \text{ , } t \geqslant T \text{ .} \tag{4}$$

The payoff for the policy maker is then

$$L_o \triangleq \int_o^\infty G(y_t - \tilde{y}_t, t)\, dt \; . \tag{5}$$

Condition (4) implies at once that

$$\lim_{t\to\infty} G(y_t - \tilde{y}_t, t) = 0 \quad \text{if} \lim_{t\to\infty} (y_t - \tilde{y}_t) = 0 \tag{6}$$

Other consequences will be discussed later. We note here, however, that as long as (4) holds the precise forms of $I(\cdot)$, $U(\cdot)$, $G_o(\cdot)$ and $G(\cdot)$ are not essential to what follows.

In order to allow for an upward trend of target demand level, $\tilde{y}_t$, while imposing a growth limit upon it, we postulate the simple form

$$\tilde{y}_t = y^\infty(1-e^{-at}) \; , \; a \text{ and } y^\infty = \text{constant} > 0 \; . \tag{7}$$

For the determination of aggregate demand, y_t, we postulate the dynamic relation

$$\dot{y}_t = - a\, y_t + k_\mu\, \mu_t + r_g\, g_t + r_y \; , \; y_o \text{ given} \tag{8}$$

where

μ_t is an index of the "tightness" of the money market that can be controlled by the government as discussed below,

g_t is an index of the government's fiscal policy,

k_μ is the "policy multiplier" for the monetary policy index, k_μ = constant > 0 ,

r_g is the "policy multiplier" for the fiscal policy index, $r_g > 0$,

r_y is a "forcing term".

The policy multiplier r_g and the forcing term r_y may depend, in an unknown fashion, on μ_t , $y_t - \tilde{y}_t$ and t .

Equation (8) requires some explanation. On combining (7) and (8) one has

$$\frac{d}{dt}(y_t - \tilde{y}_t) = - a(y_t - \tilde{y}_t) + (r_y - a\, y^\infty) + k_\mu\, \mu_t + r_g\, g_t \; . \tag{9}$$

If there are no external disturbances, $r_y - a\, y^\infty$, and no effects of monetary policy, $k_\mu\, \mu_t$, and fiscal policy, $r_g\, g_t$, the "nominal system"

$$\frac{d}{dt}(y_t - \tilde{y}_t) = - a(y_t - \tilde{y}_t)$$

describes the situation. Its response is stable at an exponential decline rate.

The term $r_y - a\, y^\infty$ may represent any stimulating or depressing force acting on the aggregate demand rate; it may be of an exogenous nature or of the induced, expectational variety. We suppose only that this term has known bounds; that is, $\underline{r}_y \leq r_y \leq \bar{r}_y$, where $\underline{r}_y$ and $\bar{r}_y$ are given constants.

Monetary policy is usually carried out by a monetary authority either by open market operations in the bond market or by setting the rediscounting rate. Massive and sustained applications of such measures will affect the lending operations of the commercial banks and hence the aggregate demand. The crux of the matter is the delay in response. To reflect this uncertainty we postulate that the aggregate demand responds to a "smoothed" monetary policy index, μ_t , rather than to the current monetary policy indicator, m_t , itself. We allow the smoothing coefficient, r_μ , to be an unknown function of μ_t , $y_t - \tilde{y}_t$ and t . We suppose only that its bounds are known; that is, $\underline{r}_\mu \leqslant r_\mu \leqslant \bar{r}_\mu$, where $\underline{r}_\mu$ and $\bar{r}_\mu$ are given positive constants. In short

$$\mu_t = \int_{-\infty}^{t} [r_\mu \exp(-\int_{\tau}^{t} r_\mu \, ds)] \, m_\tau \, d\tau \tag{10}$$

where the coefficient of m_τ represents the weight accorded m_τ at $\tau \leqslant t$. Equivalently, we have

$$\dot{\mu}_t = r_\mu(m_t - \mu_t) \quad , \quad \mu_o \text{ given} \quad . \tag{11}$$

Thus, the present aggregate demand rate is influenced by the weighted sum of all earlier monetary controls.

In contrast, fiscal controls appear in the form of tax provisions and rates as well as in decisions about government spending. Uncertainty in the effect of fiscal controls comes from a different quarter, namely, the policy multiplier. The net effect is the product of fiscal control and its policy multiplier. The latter may depend, in some unknown way, on μ_t , $y_t - \tilde{y}_t$ and t . Here we suppose that its bounds are known; that is, $\underline{r}_g \leqslant r_g \leqslant \bar{r}_g$, where $\underline{r}_g$ and $\bar{r}_g$ are given positive constants.

Now we turn to a discussion of "control capabilities". We assume the following:

(a) The fiscal control g_t can take on any value in the interval $[\underline{g} , \bar{g}]$ where $\underline{g}$ and $\bar{g}$ are prescribed constants, $\underline{g} < 0 < \bar{g}$.

(b) The monetary control, m_t , can take on any value on the interval $[\underline{m}|\mu_t| , \bar{m}|\mu_t|]$ where $\underline{m}$ and $\bar{m}$ are prescribed constants, $\underline{m} < 0 < \bar{m}$. Thus, what constrains the actions of the monetary authority is the relative rate of expansion or contraction vis-a-vis the accumulated actions of the past, $m_t/|\mu_t|$.

Next we postulate two conditions for subsequent use.

(i) Adequacy of fiscal capability:

$$\max \{|\underline{r}_y - a\, y^\infty| \, , \, |\bar{r}_y - a\, y^\infty|\} < \underline{r}_g \min \{|\underline{g}| \, , \, \bar{g}\} \tag{12}$$

(ii) Adequacy of monetary capability:

$$\frac{\bar{r}_\mu - \underline{r}_\mu}{2\underline{r}_\mu} < \min \{|\underline{m}| , \bar{m}\} . \qquad (13)$$

Significantly, in (12), where the bounds on the "noise", $r_y - a\, y^\infty$, are compared with the bounds on fiscal capability, only the weakest policy multiplier, $\underline{r}_g$, is relevant; the value of $\bar{r}_g$ is of no consequence. In (13), what is important is the range of sudden changes in the lag-in-effect, $(\bar{r}_\mu/\underline{r}_\mu) - 1$, as compared with the range of monetary control flexibility.

III. RESULTS

We reach the following conclusions which are derived in Leitmann and Wan (1977).

1. Stabilization. If conditions (12) and (13) are met then there exist policy rules, that is, fiscal and monetary controls, which depend on the observed state μ_t and $y_t - \tilde{y}_t$ of the economy, such that the origin is uniformly asymptotically stable (Lyapunov stable) regardless of

(i) initial state $(\mu_o, y_o - \tilde{y}_o)$,

(ii) uncertainties $r_k(\mu_t, y_t - \tilde{y}_t, t)$, $k = g, y, \mu$.

In particular,

$$\lim_{t\to\infty} \mu_t = 0 \quad , \quad \lim_{t\to\infty} (y_t - \tilde{y}_t) = 0 \quad .$$

2. Welfare Loss. As noted earlier, the payoff to the policy maker, that is, the welfare loss over the infinite horizon, L_o , is given by (5). If conditions (12) and (13) are met and the consequent stabilizing policies are employed, and if the avoidable loss function, $G(\cdot)$, satisfies (4), then the welfare loss is bounded by a positive level that depends only on the initial state. Furthermore, both the welfare loss rate (avoidable loss) and the welfare loss itself go to zero as time increases. In particular, if

$$G(y_t - \tilde{y}_t, t) = (y_t - \tilde{y}_t)^2 \qquad (14)$$

so that

$$L_o = \int_o^\infty (y_t - \tilde{y}_t)^2 \, dt \qquad (15)$$

then

$$L_o \leqslant \bar{L}(\mu_o, y_o - \tilde{y}_o) \triangleq \frac{1}{2a} \left[\frac{k_\mu \mu_o^2}{c_\mu(c_\mu + a)} + \frac{2\, k_\mu\, \mu_o (y_o - \tilde{y}_o)}{c_\mu + a} + (y_o - \tilde{y}_o)^2 \right] \qquad (16)$$

where $c_\mu \triangleq \frac{1}{2} (\underline{r}_\mu + \bar{r}_\mu)$.

In general, for any admissible $G(\cdot)$,

$$\int_0^\infty G(y_t - \tilde{y}_t, t)\, dt < \infty \tag{17}$$

and

$$\lim_{t\to\infty} G(y_t - \tilde{y}_t, t) = 0 , \tag{18}$$

$$\lim_{t\to\infty} \int_t^\infty G(y_\tau - \tilde{y}_\tau, \tau)\, d\tau = 0 . \tag{19}$$

3. Policy Structure. We must distinguish two cases.

(a) If (12) is met but (13) is not, stability and bounded welfare loss can be assured by the alternative policy of using only fiscal control, namely,

$$\begin{aligned} m_t &= 0 \ \text{ and } \ g_t = \underline{g} \ \text{ if } \ y_t - \tilde{y}_t > 0 , \\ m_t &= 0 \ \text{ and } \ g_t = \bar{g} \ \text{ if } \ y_t - \tilde{y}_t < 0 . \end{aligned} \tag{20}$$

In view of (11), if $m_t \equiv 0$, $|\mu_t|$ decays and so, by (12), a time is reached eventually such that thereafter

$$\max \{|\underline{r}_y - a\, y^\infty + k_\mu\, \mu_t| , |\bar{r}_y - a\, y^\infty + k_\mu\, \mu_t|\} \leqslant \underline{r}_g \min \{|\underline{g}| , \bar{g}\} . \tag{21}$$

This suffices to assure asymptotic stability by use of the alternative policy.

(b) If (12) and (13) are satisfied, both fiscal and monetary controls should be employed. The policy structure in feedback form is shown in Figure 2.

4. On Macro-economic Debates. If, as we have done, one can characterize monetary control as control with uncertain lag and fiscal control as control with uncertain effectiveness, then fiscal control is more fundamental in the sense that stability can be assured or not depending on, and only on, the adequacy of fiscal capability. On the other hand, given the assured stabilizability of the system, monetary neutralism is justified or not depending on the adequacy of monetary capability (that is, on the predictability of the delay in effect of monetary control relative to the flexibility available to the monetary authority in pursuing expansive or contracting policies).

5. State Space Behavior. The behavior of the trajectories of the stabilized system "near" the origin (that is, after a sufficiently long time) is discussed in Leitmann and Wan (1977).

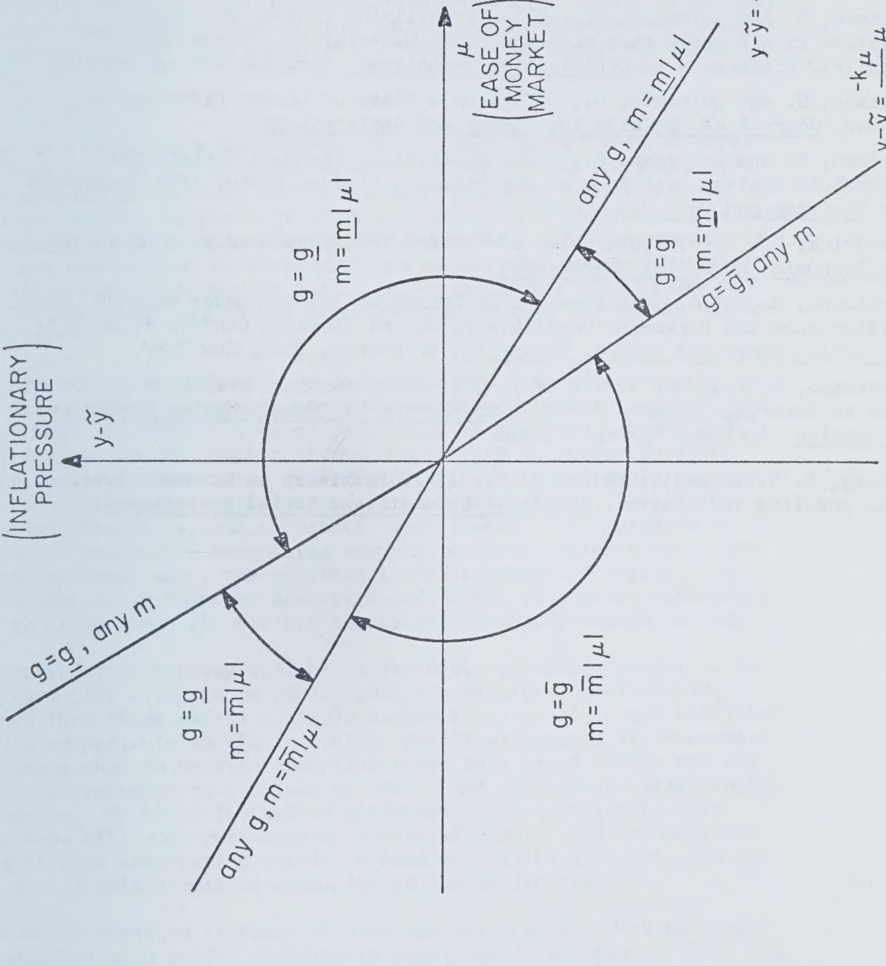

(INFLATIONARY PRESSURE)
$y-\tilde{y}$
μ
(EASE OF MONEY MARKET)
$g=\underline{g}$, $m=\underline{m}|\mu|$
any g, $m=\underline{m}|\mu|$
$g=\bar{g}$, $m=\underline{m}|\mu|$
$g=\bar{g}$, any m
$y-\tilde{y}=\frac{-k_{\mu}}{c_{\mu}+a}\mu$
$y-\tilde{y}=\frac{-k_{\mu}}{c_{\mu}}\mu$
$g=\underline{g}$, any m
$g=\underline{g}$, $m=\bar{m}|\mu|$
any g, $m=\bar{m}|\mu|$
$g=\bar{g}$, $m=\bar{m}|\mu|$

Figure 2. Policy Structure

REFERENCES

Friedman, M., 1956, Studies in the Quantity of Money, University of Chicago Press, Chicago.

Friedman, M., 1959, Program for Monetary Stability, Fordham University Press, New York.

Friedman, M. and Schwartz, A., 1963, Money and Business Cycles, Review of Economic Statistics, Supplement.

Gutman, S., 1976, Uncertain Dynamical Systems - A Differential Game Approach, NASA TMX-73, 135.

Gutman, S. and Leitmann, G., 1975a, Stabilizing Control for Linear Systems with Bounded Parameter and Input Uncertainty, Proceedings of 7th IFIP Conference on Optimization Techniques, Springer Verlag, Berlin.

Gutman, S. and Leitmann, G., 1975b, On a Class of Linear Differential Games, Journal of Optimization Theory and Applications.

Gutman, S. and Leitmann, G., 1976, Stabilizing Feedback Control for Dynamical Systems with Bounded Uncertainty, Proceedings of IEEE Conference on Decision and Control.

Kendrick, D., 1976, Application of Control Theory to Macroeconomics, Annals of Economic and Social Measurement.

Leitmann, G., 1976, Stabilization of Dynamical Systems under Bounded Input Disturbance and Parameter Uncertainty, Second Kingston Conference on Differential Games and Control Theory II, M. Dekker, Inc., New York.

Leitmann, G. and Wan, H. Y., Jr., 1977, Macro-economic Stabilization Policy for an Uncertain Dynamic Economy, New Trends in Dynamic System Theory and Economics, Springer Verlag, Vienna.

Chang, S. S. L. and Steckler, H. O., 1977, Fuzziness in Economic Systems, Its Modeling and Control, Annals of Economic and Social Measurement.

ON THE CONCEPTUAL FOUNDATIONS OF NORMATIVE "PUBLIC POLICY ANALYSIS": AN INTERDISCIPLINARY INTERPRETATION AND SYNTHESIS

HORACE W. BROCK
Stanford Research Institute
Menlo Park, California 94025

Abstract: The author draws upon cooperative game theory to establish some interrelationships between simple models of ethics, politics, and economics. The unified perspective introduced is intended to bring us a step closer to answering one of the root questions of public policy analysis: namely that of the proper "bottom line" for evaluating Government policy decisions. The author also shows that our ostensibly incompatible concerns for equality and for optimality are in fact compatible. They are mathematically dual to one another.

To what criterion can a Government administrator appeal when he seeks to make a "rational" or "fair" decision which is "in the public interest"? What do the adjectives rational and fair mean in a public policy context where there are several people with partially conflicting interests at stake? Not until we have a better understanding of and indeed tentative answers to these questions can be come to terms with one of the most pressing problems in the new discipline of normative public policy analysis: the nature of the proper "bottom line", or to use more classical terms, The Social Welfare Function (SWF).

Three different disciplines have contributed to such understanding as we have of this difficult problem: moral philosophy, political science, and economics. These disciplines have given us what we might call three different perspectives on the problem of the bottom line in policy analysis. Moral theorists have investigated criteria of distributional equity and/or of the nature of "the public good" taken in an ethical sense. Political scientists--and especially game theorists--have looked at the problem as one of determining a reasonable bargaining compromise which reflects the underlying balance of power among the possible interest groups. Finally, economists have taught us to think of decentralized market allocation mechanisms which tend to bring about (in certain conditions) a supply/demand balance.

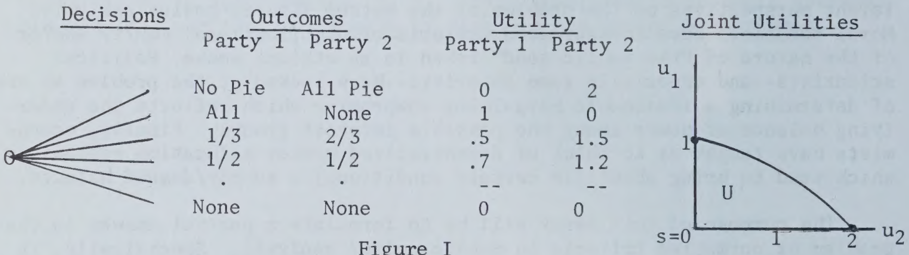
The purpose of this essay will be to formulate a partial answer to the problem of normative criteria in public policy analysis. Specifically, it will be shown that there exists a single mathematical model, which underlies theoretically respectable models in ethics, political science and economics. It will be shown that in certain important cases this model yields one and the same answer (outcome) to problems in ethics and in politics (bargaining) and in economics. To state the matter differently the solution to this model is at once ethically respectable, political stable, and economically optimal. With this interesting result in hand we will be in a position to defend the use of this normative model for policy applications.

The model in question is known to game theorists as the Nash-Harsanyi-Shapley "Value" of an n-person cooperative game. In this paper we shall not review the game theory proper in much detail. What we shall do is to show that in light of certain important recent results, the Value concept can be used to build bridges between the disciplines of ethics, politics, and economics. For simplicity we shall commence with two-person problems and consider the relationship between an ethical model and a political model. Thereafter we consider $n > 2$ and deal with all three disciplines.

There is a secondary purpose to this essay which should be pointed out at the outset. No matter whether a decision problem be ethical or political in nature, there are essentially two different perspectives from which we appraise a "solution" to the problem. First there is what we shall call the equalization perspective. Here the emphasis is on what should be made to be equal to what. Second there is the optimization perspective: what do we wish to maximize or minimize? The game theoretic formalism we introduce will be shown to encompass both perspectives simultaneously.

I. Two-Person Problems in Ethics and Politics

I.A. Ethics: Consider a prototypical pie division problem between two people. More specifically suppose that a parent must decide upon which division of a given pie between his two children is most "fair" in an ethical sense. Assume that the parent knows the (cardinal) utility functions of the two children. Moreover assume that the parent has sufficient knowledge to interpersonally calibrate the two utility scales in the following sense: a distribution of pie which awards a utility increment of x to party one and 2x to party two implies that party two's utility gain (i.e. welfare gain) is twice that of party one. The reader concerned by our introduction of interpersonal comparisons will appreciate that there is no ostensible escape from such comparisons when an ethical problem is being analyzed (but see Theorem I below). As is well-known interpersonal comparisons are neither necessary nor desirable when many political and/or economic problems are analyzed. The pie division problem is summarized schematically in Figure I. The "prospect space" diagram on the far right summarizes all of the ethically pertinent information about the decision problem. [This assertion is equivalent to one version of the Individualism Postulate of modern welfare economics and ethics (e.g. Harsanyi [1955]).]



Decisions	Outcomes		Utility	
	Party 1	Party 2	Party 1	Party 2
	No Pie	All Pie	0	2
	All	None	1	0
	1/3	2/3	.5	1.5
	1/2	1/2	.7	1.2
	.	.	--	--
	None	None	0	0

Figure I

We shall now sketch a new solution theory for this canonical fair division problem. What is needed of course is a morally appealing criterion of equitable distribution. But distribution of what? Pie? No, only in an ancillary sense. For a little thought suggests that what we ultimately wish to distribute fairly is utility (welfare). Pie itself is only important because it generates utility. [It is for this reason that the abstract SWF of the Bergson-Samuelson-Arrow school is denoted by $W(\underline{u})$. Emphasis properly is on the individual utility levels.]

For a solution to the pie division problem we shall propose a version of the familiar maxim "To Each According to His (Relative) Need" which does justice to this last observation. The proposed solution must satisfy the following two conditions:

Condition (i): The outcome must be Pareto efficient;
Condition (ii): The outcome must satisfy the following equation

which characterizes distribution of utility in accord with relative need[1]:

$$-\frac{u_1-s_1}{u_2-s_2} = \frac{du_1}{du_2} \triangleq \mathrm{RN} \qquad (1)$$

where the vector s designates the status quo (no pie) payoff; where du_1/du_2 denotes the slope of the boundary of the prospect space in Figure I, a slope which corresponds to a (local) measure of relative need (RN) provided we are willing to identify relative need with relative intensity of desire for pie; where s is the disagreement payoff ("No Pie").

The reader may verify immediately that there is only one point on the boundary of the prospect space (i.e. one Pareto efficient point) that satisfies this relation: this is the point defined by the intersection of the two lines $\overline{AB}$ and $\overline{CD}$ shown in Figure II. The lines are constructed to have the property that their slopes are equal in magnitude, but opposite in sign[2].

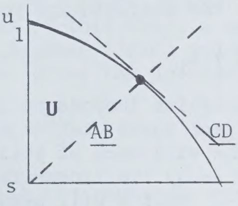
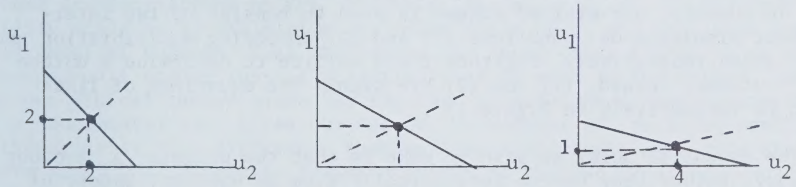


Figure II

Is the distribution of welfare implied by this scheme "reasonable"? One way to answer this question is to consider the implications of the norm in the setting of some simple examples. We attend to this in Figure III. For simplicity, the boundaries of the prospect spaces here are all "flat", corresponding to cases where there exists a global measure of relative need. In Case A, where the two parties desire the pie equally, utility is split 50-50 which is reasonable. In Case C party two receives four times the payoff that party one receives. Why a 4:1 ratio? Because (by construction) he has four times the need for the pie. This seems reasonable to us, but the reader can judge for himself.



Case A Case B Case C

Figure III

---

[1] There is of course an alternative distributive norm, "To Each According to his Contribution". But this does not apply to the case in question since -- by assumption -- the two children have not contributed to baking the pie.

[2] To the best of our knowledge, the norm we are proposing is perhaps the only unambiguous and well-defined "relative needs" norm proposed thusfar.

Let us pause momentarily to observe the important duality between "equalization" and "optimization" that is embedded in the distributive norm we have set forth. The concern of the "fair-sharing" norm (1) is clearly with that which should be equalized across parties, -- i.e. with "equilibrium". Or to state the matter metaethically, (1) is a deontological as opposed to a teleological assertion. We can perceive this concern with equalization better, perhaps, if we transpose (1) and write

$$a_1(u_1-s_1) = a_2(u_2-s_2) \qquad (1^*)$$

where the variables a_i are simply the slope variables (e.g. the direction cosines) of Figure II[1] above. (1*) says each person will get the same amount of utility once the utilities are weighted by the relative needs. Now for certain policy applications, as well as for philosophical purposes, we might wish to think of ethics not in terms of equalization, but in terms of maximization of social welfare, or something such. Can we interpret our solution in this mode? Yes, the solution characterized by Conditions (i) and (ii) above will clearly solve the maximization problem

$$\max_{U} \sum_{i} a_i u_i \qquad (i = 1,2) \qquad (2)$$

where the a_i are the slope variables introduced in (1*), and where U denotes the set of points in the prospect space. Let us be clear on what is being asserted here. Technically, the existence of joint strategies (including joint randomizations) implies -- by the linearity of von Neumann-Morgenstern utility functions in utility -- that U will always be convex. Hence any efficient point will maximize some weighted sum-of-the-utilities. Thus a question that a policy analyst can and should come to terms with is very simply: How much weight should be given to whose interests, and why?

The scheme we have sketched above provides an unambiguous answer to this. What is maximized is the sum of the needs-weighted utilities. But we are now in a position to observe the duality we have mentioned above. Note that just as we could not determine the "proper" weights a_i in (2) without invoking the distributive norm (1), we likewise could not solve (1) itself without laying down (recall Condition (i) above) a "group maximization" or efficiency condition on top of the distributional condition. And note that equation (2) is none other than a prescription for an efficient outcome! In summary, our ethical scheme is seen to consist of two interdependent simultaneous equations (1) and (2) embodying equilibration and optimization respectively. Together these suffice to determine a unique "fair" outcome. Indeed, (1) and (2) are simply the equations of lines AB and CD respectively in Figure II[1].

The essential point we wish to make is that the weights a_i are not arbitrary. Rather they emerge intrinsically from an explicit theory of equity, in conjunction with the data contained in the pair of calibrated utility scales. All this should be of interest to scholars (e.g. Keeney and Raiffa [1976]) who are interested in extending "multi-attribute decision theory" to group situations where the different parties are the attributes. But note that a linear (not a multilinear) form seems sufficient for this.

---

1. The duality relations explored here have been discussed from two different vantage points by Brock [1977] and Shapley [1969]. It should also be pointed out that there are alternative theories of distribution than our own. Harsanyi [1955] argues that unweighted utilities should be added.

I.B Politics: We are now in a position to commence with some interdisciplinary bridge-building. Suppose that a Government policy maker views his role as political rather than ethical in nature. That is, he may view his job as that of determining a policy which is most acceptable in the political sense of being the most reasonable possible compromise between competing interests. How might he proceed in this case? One approach might be to appeal to game theoretic models of rational bargaining -- and to arbitrate his decision in accord with the norms of these models. His decision would then correspond to a "rational compromise" between the parties whose interests are at stake. What does game theory tell him to do in such a situation? To answer this, we shall suppose that there are only two parties involved. I.e. n = 2; moreover we assume that there is a "natural", exogenously given "disagreement payoff" in the underlying bargaining situation. This "status quo" payoff, as it is sometimes called, will be denoted by the utility vector s. [The reader will observe that the problem just posed is isomorphic to the two-person pie division problem of Section I.A. in all important respects.] It is well-known in game theory that there is a unique solution to problems of this kind, a solution which satisfies the axioms of bargaining both of Nash [1950] and Harsanyi [1961]. The solution u* is the payoff vector which solves the problem:

$$\max_{U} \prod_{i} (u_i - s_i) \tag{3}$$

The coordinates of u* define the rectangle with the greatest area in the prospect space U. (3) is known as the Nash-Harsanyi solution. We can now state the first of several equivalence theorems that we shall use in interdisciplinary bridge-building:

Theorem I (Brock [1977]): The Nash-Harsanyi theory and the ethical theory introduced in Section I.A. are fundamentally different, as reflected by the important fact that they have different symmetry groups. Nonetheless, they are solution equivalent in the sense that the physical solutions prescribed by each (i.e. the division of pie) coincide.

This result requires interpretation. As is well known in game theory, the Nash-Harsanyi theory is invariant under the group of coordinate-wise positive linear transformations of the utilities. For example, we can multiply one utility scale by x, and divide the other scale by y, and this will not affect this physical outcome. What this means -- among other things -- is that interpersonal comparisons of utilities play no role in the theory. Yet in our own ethical theory above, we saw that interpersonal comparisons entered in a fundamental way, given our use of the concept of "relative need", etc. Mathematically this difference that we have noted translates into statements about the respective symmetry groups of the two theories. Clearly, the symmetry group of the ethical theory is a subgroup of that of the Nash theory. However, it can be shown that the first-order (maximization) conditions of the Nash-Harsanyi maximization problem (3) coincide in form with the equations (1) and (2) used to characterize an ethically fair outcome above. This is of course not at all obvious, and the reader will find a proof in Brock [1977] who builds on an observation of Harsanyi [1963]. One interpretation of all this is that an administrator who is trying to act "fairly" in the sense of Section I.A. need not worry about interpersonal calibration! For the (physical) outcome will not depend upon such calibrations.

It might seem strange that rational bargaining and ethical arbitration are solution equivalent. Yet it is not. First, bargaining by its very nature tests the relative intensity of desire of the players (Shapley [1969]).

Second, in formulating the above bargaining problem, we in effect ruled out the use of threats by the players when we assumed the existence of the vector s. In more general ("variable threat") situations, the vector s is replaced by an optimal threat payoff vector t* (Nash [1953]). Now whereas it seems unobjectionable to make use of a "natural" reference outcome like our vector s (no pie for either player) in the context of an ethical arbitration scheme, it will generally be regarded as morally inadmissible to include threats in ethical theory. We are thus led to distinguish between two classes of decision problems which we shall call s-games and t-games. Only when a given problem is an s-game can we assert equivalence between the ethical perspective of Section I.A. and the political perspective of rational bargaining. [s-games are sometimes known as "pure" bargaining games.] One final note on our equivalence result. The reader will observe that the Nash-Harsanyi theory is expressed in terms of a maximization problem (3). Yet from Theorem I we know that we can restate the theory in an equalization form (recall (1)). This can be viewed as another duality.

We can summarize our principal results thusfar as follows. Regardless of whether we interpret "acting in the 'public interest'" from a maximization or an equalization perspective, or from an ethical or a political vantage point, we have characterized a class of problems (s-games) where one and the same solution satisfies the desiderata of all perspectives simultaneously. Moreover, we have seen that even though interpersonal comparisons arise at a conceptual level, they do not arise at an operational level and hence need not trouble a policy maker. A tentative conclusion of all this is that if a given decision problem can be viewed as an s-game in a natural way, then a responsible policy analyst will recommend the Nash-Harsanyi solution which has (at least) a dual source of support. In the event that the policy problem is naturally modeled as a t-game in which the parties can avail themselves of threat strategies, then the policy maker must confront the very difficult question of the proper role of Government. Is it an administrator's function to act in an ethically respectable way? Or should he seek a "rational" compromise decision, even if threats of a very ugly type are involved in defining the compromise?

II. Multi-Person Problems in Ethics, Politics and Economics

Proceeding at a much more rapid rate, we shall now survey possibilities for interdisciplinary bridge-building when $n>2$. We shall interrelate the three different disciplines, and state three further equivalence results.

II. A Ethics: Assume that we have an n-person arbitration problem and that there is a natural disagreement point s. Then in this case it is trivial to verify that our ethical equilibrium conditions [(1) and (2)] generalize directly to the n-person case. We shall not discuss this further.

II.B Politics: n-person cooperative game theory has proposed a host of different models of n-person rational bargaining. For our purposes we shall restrict our attention to two different models which capture complementary aspects of n-person theory: (i) the Harsanyi n-person bargaining model; and (ii) the Shapley "Value" of a game (Harsanyi [1963] and Shapley [1969]). Let us start with Harsanyi's scheme. Here it is straightforward to distinguish between s-games and t-games. In the n-person case, a game is an s-game if both threats and coalitions are ruled out, and there is a given "no agreement" outcome s. Harsanyi's solution in this case is

$$\max_{U} \prod_{i} (u_i - s_i) \qquad i = 1,..,n \tag{4}$$

which is clearly identical to the two-person case (3). In the case of a t-game, Harsanyi's solution can be summarized by the Nash-like form

$$\max_{U} \prod_{i} (u_i - t_i^*) \qquad i = 1,\ldots,n \tag{5}$$

where the vector t* is a "net threat" payoff vector whose i^{th} component is the net threat payoff to player i from all coalitions he might join. [In Harsanyi's model, t* and the solution u* are determined simultaneously.]

We have seen that just as in the two-person case, if an n-person policy problem can reasonably be modeled as an s-game, then the ethical solution and the political solution will coincide. This will not be true in the event the problem is a t-game, in which case the policy maker will have to confront the problem of the proper function of Government.

The second n-person model we shall discuss is the Shapley Value. The Value is an axiomatized solution concept (not necessarily unique) which awards each person a utility payoff equal to his average 'marginal contribution' to the coalitions which he might join. We can see this formally:

$$u_i = \sum_{S} \nabla \, [v(S) - v(S - \{i\})] \qquad S \subset N \tag{6}$$

where N is the player set; S is a representative coalition; v(S) is the so-called characteristic function which denotes the "worth" (total utility) of coalition S when it opposes coalition N-S optimally; where ∇ is the generalized expectation operator $\nabla = [(s-1)!(n-s)!/n!]$ indicating that in computing the averaged marginal contribution of i to S, all orders of coalition formation are equally likely; and where s is the cardinality of S, and likewise n of N. The flavor of Shapley's scheme is quite "economic" with its emphasis on marginal contribution. And whereas on first sight his solution might appear unrelated to Harsanyi's, we can state:

Theorem II (Harsanyi [1963]): A solution to Harsanyi's bargaining model will always be a Shapley Value of the game.

This remarkable result is of special interest to us in the present paper for two reasons. First, it permits us to build bridges between ethics, politics and economics -- as we shall see in the next section. Second, it illuminates the 'duality' with which we are concerned in a new way: a scheme concerned with equalization ("To Each According to his Contribution") is equivalent to a maximization problem [recall equation (5) and our earlier discussion]. We can now turn to our final case of interest.

II.C. Economics: One of the fundamental concepts of economics is that of supply-demand balance. This "equilibrium" concept is elegantly embedded in the notion of a competitive equilibrium. To solve for a competitive equilibrium is to solve for a doublet $< x^*, p^* >$ where x* is a m-vector of commodities and p* is a m-vector of prices such that if all consumers and producers take p* "as given", then supply equals demand across all m markets. The basic normative significance of a competitive equilibrium is that it is (Arrow [1951]) Pareto efficient. Specifically, the market-clearing commodity allocation will solve

$$\max_{U} \sum_{i} \lambda_i u_i \tag{7}$$

for some set of utility weights λ_i. Now as is well-known, while a market equilibrium may be efficient, there is no reason to suppose that it is equitable in any clear sense. Or, to state the matter differently, if the

u_i -scales in (7) were the "true" interpersonally calibrated scales, then there would prima facie be nothing special about the weights λ_i in (7). We shall now draw on recent results to suggest that this is not the case. First, we can state:

Theorem III (Aumann [1975]): An allocation awarded by a competitive equilibrium in an exchange economy will coincide with the allocation awarded by the Shapley Value of the bartering game that can be identified with the market. [Note: "allocation" refers to a bundle of market commodities.]

This great correspondence theorem links the realms of politics (game theory) and economics. It tells us first of all that the market awards people a payoff equal to their marginal contribution--in utility terms, not just (as has been long-known) in dollar terms--in light of the definition of the Shapley Value[1]. Second, and more importantly for our purposes, we know from Theorem II that the Shapley Value and the Harsanyi bargaining model are equivalent. Moreover, we know from Theorem I and the subsequent discussion that if a game is an s-game, then the solution to the Harsanyi bargaining model will satisfy our ethical criteria of equitable distribution. But a market game is an s-game, since as is well known there is no overt role for threats. Hence, we can infer from this chain of reasoning that:

Theorem IV: The λ_i weights appearing in (7) are not arbitrary: rather, assuming once again that the true scales are used, the weights will serve as measures of relative need.

The reader may wonder why these assertions about market equilibria are pertinent to one of the principal topics of this paper: the bottom line in public policy analysis. The reason is that there are many people who for whatever reason champion use of the market as a means of resource allocation. Presumably the reason they place faith in the market is because of the results it yields. Hopefully, our discussion above will serve to increase our understanding of exactly what it is that the market does achieve. Consider now the role of the policy analyst who -- like many others -- may be attracted to competitive market outcomes and who might seek to apply strictly economic models to his problems. Policy analysts in Government largely make decisions concerning non-market commodities, i.e. "externalities". It is well known that if there are significant externalities in a market, there will be no price-quantity equilibrium (except in a Lindhal sense which is weak). In particular, price systems will not be well-defined. But what the foregoing discussion implies is that a policy maker does not need to turn to the apparatus of economic analysis proper in order to obtain desirable outcomes. He can bypass economics, whose models are ill-defined in the problems he is charged with, and turn directly to game theoretic models. And these latter models will of course be well-defined in environments with externalities as well as in those without externalities.

III. Conclusion

We have attempted to demonstrate that there is an intimate relationship between certain models which lie at the heart of ethics, politics, and economics. Assuming that a policy problem is characterized by the existence of a natural status quo (reference) point -- one that is not determined by active threats -- then the "outcomes" prescribed by the models of these three very different disciplines will coincide. This is significant for normative public policy analysis because it suggests that the Nash-Harsanyi-Shapley theory should be used in applications, assuming that the theory can be operationalized in a convincing manner. A policy analyst invoking the theory in an s-game situation can defend his decision to do so by appealing to the normative criteria of no less than three disciplines! On a somewhat more abstract level, we have noted the existence of an interesting set of duality relationships which may serve to reconcile widely different conceptions of the nature of the "public interest". [Duality relationships somewhat similar to those that we (and others) have noted in the social sciences abound in physics, where scholars have long been interested in restating equilibrium relationships in variational form.]

---

[1] Aumann in his paper does not give a clear interpretation of the meaning of his theorem. Elsewhere, the present author (Brock [1977]) has tried to provide a satisfactory interpretation. The difficulty at issue is that of how to interpret the "weights" associated with the generalized Shapley Value.

Several research problems are posed by the results we have reported, and by the interpretations we have ascribed to them. First, we shall only be able to establish Theorem IV in a rigorous manner when we understand the exact relationship between two different "generalized Shapley Values" which we have taken the liberty to interchange in the foregoing discussion: that of Harsanyi [1963], and that of Shapley [1969]. Second, there is the troublesome problem of non-uniqueness of the solution concepts we have used throughout when $n>2$. This is clearly an important problem if normative applications are intended -- for a policy analyst must recommend some one course of action as optimal. In this respect we note the recent progress of Harsanyi [1977] in identifying a unique solution. Third, research is needed that will facilitate operationalization of game theoretic models, especially when n is "large". See Owen [1972] for impressive results in this direction.

Of particular interest and concern to the present author is the somewhat arbitrary distinction between an s-game and a t-game that we have introduced. Recall why this is so important. According to the theory we have advanced, if a decision problem is an s-game, then one and the same outcome will be both politically acceptable and ethically fair. But if a game is a t-game, then a policy maker will be forced to choose between two roles: that of an ethical arbitrator, and that of a "politician". Now suppose that a problem is a t-game and that a policy maker wishes to adopt the fair course of action. Our theory (unlike, say, the Utilitarian theory) requires that he identify (invent?) some point s in order that the fair outcome be well-defined. How will he choose s? This is clearly a deep problem. Even in the case of a true s-game, the ethical status of s itself is not fully clear. Suppose that s corresponds to people's initial endowments in the marketplace. We know by Theorems III and IV that the solution determined by the Nash-Harsanyi-Shapley theory will be equitable relative to s. But suppose that s corresponds to a social state characterized by tremendous inequalities in endowments? What then of our "solution"?

We feel that the problems outlined in the preceeding paragraph can be overcome. In this concluding paragraph, we shall sketch how. Clearly none of the problems cited just above would arise if it could be agreed upon that the proper role of policy makers is political -- not ethical. For in this event, the whole notion of an ethically respectable status quo point s would not arise. We happen to feel that true social justice does entail political behavior on the part of policy makers. How so? Neither individual decisions nor public laws and policies are made in an ethical vacuum. For in general, the deepest ethical convictions of the people will be embodied in a Constitution of the land. Elsewhere (Brock [1977]) we have set forth a comprehensive theory of justice in which a case is made for highly "ethical" behavior when a Constitution is chosen, and "political" behavior in all subsequent decisions. These subsequent decisions include public policies, as well as the "rational life plans" of all individuals.

Acknowledgements: The author is indebted to discussions with Kenneth Arrow, Robert Aumann, John Harsanyi, and Lloyd Shapley. This research was supported in part by the Andrew Mellon Foundation, the Aspen Institute of Humanistic Studies, the National Science Foundation (NSF Contract No. C-915), and the U.S. Energy Research and Development Administration (ERDA Contract MEU-5034) where Mr. Lester Fink has provided important support for this work.

Bibliography: List of References Cited

Arrow, Kenneth J., 1951, "An Extension of the Basic Theorems of Welfare Economics", *Proc. of* 2^{nd} *Berkeley Symp. on Math. Stats. and Probability.*

Aumann, Robert, 1975, "Values of Economies with a Continuum of Traders", *Econometrica,* Vol. 43, No. 4.

Brock, Horace W., 1977, "A Game Theoretic Account of Social Justice", *Proc. of the Math. Soc. Sci. Board Conf. on Game Theory and Pol. Sci.*

Harsanyi, John C., 1955, "Cardinal Welfare, Individualistic Ethics and Interpersonal Comparisons of Utility", *Journal of Political Economy,* 63.

Harsanyi, John C., 1961, "On the Rationality Postulates Underlying the Theory of Coop. Games", *Journal of Conflict Resolution,* 5.

Harsanyi, John C., 1963, "A Simplified Bargaining Model for the n-Person Cooperative Game", *International Economic Review,* 4.

Harsanyi, John C., 1977, "A Solution Theory for Noncooperative Games and its Implications for Coop. Games", *Proceedings of the Mathematics Social Science Board Conference on Game Theory and Political Science.*

Keeney, Ralph, and Howard Raiffa, 1976, *Decisions with Multiple Objectives,* John Wiley and Sons.

Nash, John F., 1950, "The Bargaining Problem", *Econometrica,* 18.

Nash, John F., 1953, "Two-Person Cooperative Games", *Econometrica,* 21.

Owen, Guillermo, 1972, "Multilinear Extensions of Games", *Managemt. Sci.*,18.

Shapley, Lloyd S., 1953, "A Value for n-Person Games", *Annals of Mathematics Studies,* No. 28.

Shapley, Lloyd S., 1969, "Utility Comparisons and the Theory of Games", appearing in *La Decision II: Aggregation et Dynamique des Ordres de Preference,* Ed. by G. T. Guibald, Paris: Centre National de la Recherche Scientifique.

BY-PASSING THE FREQUENCY DOMAIN

YEHONATHAN HAZONY
Princeton University, Princeton, New Jersey

DOV HAZONY
Case Western Reserve University, Cleveland, Ohio

ABSTRACT

Recent developments in the methodology of interactive computer graphics on large scale time-sharing computers provide very fast computerized tools for circuit design and analysis. These tools enhance the attractiveness of the State Variable approach to the problem. In this paper, the impulse response of RLC networks is derived in the time domain. Among the various possible numerical algorithms, the one chosen is based on an explicit n^{th} order time predictor, which takes on a Taylor, Chebychev, or intermediate polynomial forms, depending on the 'stiffness' of the system. An implementation in APL-Graphics provides an instant numerical as well as pictorial description of the propagation of signals throughout the system.

INTRODUCTION

The application of the digital computer to electrical engineering has had a profound effect on the methods of electronic circuit design and analysis. The discipline of computer aided design (CAD) which has reached wide acceptance and maturity, has been recently described in detail by Jensen and McNamee [1]. Most of the work has been done in the environment of the batch computer at the expense of inherent human inefficiencies.

The advent of interactive computer graphics on a large scale main frame computer in the virtual machine environment shifts the emphasis from machine efficiencies to human efficiencies, providing for very fast computerized tools for circuit design and analysis. A primary advantage of the fast response of the interactive system is in the prominent role which could be played by human intuition on all levels of the design process. As reviewed by Jensen and McNamee [1] there are many ways whereby the computer is used in the design, though in this paper the emphasis is on State Variables methods [2].

An important mode of electronic data transmission and processing is based on the use of sequential slices (slots) in the time domain, each being dedicated to an entirely independent piece of information. It is imperative that any signal processing in a given time slice will not affect or be affected by the information contained in the adjacent time slices. Time limited RLC filters are designed to adhere to the 'absolute' separation between time slices.

The traditional method of analysis of RLC network, Laplace transforms the problem from the time to the frequency domain and as such is inappropriate for the study of time limited filters. In this method a square pulse drive is transformed into a finite frequency distribution, where the intrinsic errors, due to the finiteness of the distribution, are spread out over the entire time domain. Such errors compound the problem since they are introduced by the analysis rather than the network.

Alternatively, the frequency domain may be by-passed and the entire analysis is performed in the time domain. The method entails the description of the RLC network in terms of a set of state differential equations, representing the time evolution of the input signal as a boundary value problem. The present paper reviews the procedure for setting up the state equations, and presents a simple numerical algorithm for the determination of the state variables. These procedures are implemented in an interactive computing environment using APL-Graphics, producing numerical as well as pictorial presentations of the results of the analysis. The computer system provides for instantaneous display of the results of the analysis.

NETWORK REPRESENTATION

Kirschof's law for a single loop network may be written as

$$V - Sq - Ri - L\frac{di}{dt} = 0 \tag{1}$$

or in a differentiated form

$$\frac{dV}{dt} - Si - R\frac{di}{dt} = L\frac{d^2i}{dt^2} \quad . \tag{2}$$

This expression is valid for a multi-loop network with V and i represented as time dependent vectors and S, R and L denoting the capacitative, resistive and inductive impedence matrices. The order of this set of equations is reduced by

$$\frac{dV}{dt} - Si - RI = L\frac{dI}{dt}$$

$$I = \frac{di}{dt} \quad , \tag{3}$$

yielding a system of 2n first-order equations for an n-loop network. In the case of a single input to the network, the V vector contains a preset sequence of values pertaining only to the equation describing the input loop. The above set of equations may be streamlined by denoting the state variables vector J to include 2n + 1 elements,

$$J_o = \frac{dV}{dt} \quad ,$$

$$J_{1,2,\ldots,n} = i_{1,2,\ldots,n}$$

$$J_{n+1,\ldots,2n} = \frac{d}{dt}\, i_{1,2,\ldots,n} \quad , \tag{4}$$

with the boundary values given by J_o.

The n-loop two-port network is represented by

$$A \frac{dJ}{dt} = BJ \tag{5}$$

or

$$\frac{dJ}{dt} = CJ \tag{6}$$

with

$$C = A^{-1} \cdot B \quad . \tag{7}$$

Example 1: Consider the two-loop network

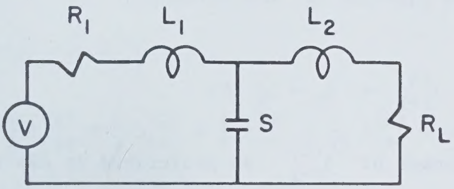


Figure 1: A Simple RLC Network

The respective matrices are given by

$$A = \left(\begin{array}{ccc|cc} 1 & 0 & 0 & & \\ 0 & 1 & 0 & & 0 \\ 0 & 0 & 1 & & \\ \hline & & & L_1 & 0 \\ & 0 & & & \\ & & & 0 & L_2 \end{array}\right)$$

$$B = \left(\begin{array}{ccc|cc} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ \hline 1 & -S & S & -R_1 & 0 \\ 0 & S & -S & 0 & -R_L \end{array}\right)$$

and the C matrix is given by Equation 7. The diagonality of the A matrix reflects the simplicity of the particular network, while more complicated networks will involve non-diagonal elements.

A NUMERICAL REALIZATION

A numerical realization of an input signal propagating in a given network, entails an algorithm which generates the time-evolution of all state variables given the A and B matrices and the time dependence of the input signal. A polynomial numerical time predictor may be constructed for that purpose [3]. Using a discrete time scale with constant increments, $t_{\ell+1} = t_\ell + \Delta t$, one could estimate the state vector at time $\ell+1$ in terms of J_ℓ and the difference equation

$$\Delta J = J'_\ell \, \Delta t = C \cdot J_\ell \, \Delta t \tag{8}$$

which is derived from Equation 6. Consequently,

$$J^{(1)}_{\ell+1} = J_\ell + J'_\ell \, \Delta t \quad , \tag{9}$$

setting the first element of $J^{(1)}_{\ell+1}$ as prescribed by the boundary condition

$$J^{(1)}_{\ell+1,0} = \frac{d}{dt} V_{\ell+1}$$

and continuing, we have

$$J^{(2)}_{\ell+1} = J_\ell + (J^{(1)}_{\ell+1})' \, \Delta t$$

which upon differentiation and substitution of Equation 9, yields

$$J^{(2)}_{\ell+1} = J_\ell + J'_\ell \, \Delta t + J''_\ell (\Delta t)^2 \quad . \tag{10}$$

Setting consistently the value of the first element of $J^{(n)}_{\ell+1,0}$ as

$$J^{(n)}_{\ell+1,0} = \left(\frac{dV}{dt}\right)_{\ell+1} \tag{11}$$

one generalizes

$$J^{(n)}_{\ell+1} = J_\ell + (J^{(n-1)}_{\ell+1})' \, \Delta t \quad . \tag{12}$$

Iterating up to $n=5$ yields the sequence

$$
\begin{aligned}
J_{\ell+1}^{(1)} &= J_\ell + J'_\ell\,\Delta t \\
J_{\ell+1}^{(2)} &\equiv J_\ell + J'_\ell\,\Delta t + J''_\ell(\Delta t)^2 \\
J_{\ell+1}^{(3)} &\equiv J_\ell + J'_\ell\,\Delta t + J''_\ell(\Delta t)^2 + J'''_\ell(\Delta t)^3 \\
J_{\ell+1}^{(4)} &\equiv J_\ell + J'_\ell\,\Delta t + J''_\ell(\Delta t)^2 + J'''_\ell(\Delta t)^3 + J_\ell^{iv}\,(\Delta t)^4 \\
J_{\ell+1}^{(5)} &\equiv J_\ell + J'_\ell\,\Delta t + J''_\ell(\Delta t)^2 + J'''_\ell(\Delta t)^3 + J_\ell^{iv}(\Delta t)^4 + J_\ell^{v}(\Delta t)^5 \quad .
\end{aligned}
\tag{13}
$$

A linear combination of $J_{\ell+1}^{(n)}$

$$J_{\ell+1} = \frac{1}{2}\,J_{\ell+1}^{(1)} + \frac{1}{3}\,J_{\ell+1}^{(2)} + \frac{1}{8}\,J_{\ell+1}^{(3)} + \frac{1}{30}\,J_{\ell+1}^{(n)} + \frac{1}{120}\,J_{\ell+1}^{(5)} \tag{14}$$

is used to generate the 5th order Taylor's predictor. Substituting Equation 13 into Equation 14, one obtains

$$
\begin{aligned}
J_{\ell+1} \equiv\; & J_\ell + J'_\ell\,\Delta t + \frac{1}{2}\,J''_\ell(\Delta t)^2 + \frac{1}{6}\,J'''_\ell\,(\Delta t)^3 \\
& + \frac{1}{24}\,J_\ell^{iv}(\Delta t)^4 + \frac{1}{120}\,J_\ell^{v}(\Delta t)^5 \quad .
\end{aligned}
\tag{15}
$$

The 5th order Taylor's time predictor has been realized numerically through Equation 9, followed by four iterations of Equation 12, and concluded by Equation 14. The boundary values are enforced through Equation 11.

STIFF NETWORKS

The particular class of filters discussed in the present sequence of papers have been designed using a synthesis method based on terminated transmission lines. This method results in very stable circuits represented by a correspondingly stable set of state equations. However, such a system could inevitably be sensitive to the performance tolerence of the physical components used in the hardware realization. In the analysis of such networks, it is important to verify the robustness of the design to imperfections in the choice of components as well as termination. Inexactness of the terminated transmission line process may give rise to high frequency near instability in the circuit, which will be represented as a high eigenvalue in the Jacobian matrix governing the state equations. Such equations are commonly described as "stiff equations", where the high eigenvalue is responsible for an instability in the numerical integration. When explicit integration procedures are pursued, one way the instability can be overcome is by reducing the Δt step size of the integration. A more efficient method involves modifying the coefficients of Equation 14 to produce a polynomial time predictor, with coefficients differing from those of the Taylor's series, but with much improved stability performance [3,4]. Several sets of such polynomial coefficients are given in Table I, providing different degrees of trade-offs between stability and accuracy.

TABLE I

AIJ^a / a_k	A50 (Taylor)	A41	A23	A14 (Chebychev)
a_1	1	1	1	1
a_2	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0.16
a_3	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{11.8}$	8.96×10^{-3}
a_4	$\frac{1}{24}$	$\frac{1}{24}$	$\frac{1}{180.25}$	2.048×10^{-4}
a_5	$\frac{1}{120}$	$\frac{1}{244.6}$	$\frac{1}{8134}$	1.6384×10^{-6}
Δt^b	.001	0.0018	0.0065	0.0155
$\lvert\lambda_{max}\Delta t\rvert^c$	3.3	6.05	19.4	50

a. In AIJ, I denotes the number of terms in the polynomial consistent with Taylor's approximation and J denotes the number of additional terms used to optimize stability.

b. Empirical values derived for the F2C 20 filter.

c. Theoretical stability limits [3,4].

These coefficients correspond to the various implementations of the polynomial time predictor

$$J_{\ell+1} = J_\ell + \sum_{k=1}^{5} (\Delta t)^k a_k \nabla^k J_\ell , \tag{16}$$

where $\nabla^k J$ represents the k^{th} time derivative of J. The corresponding coefficients to be used in Equation 14 may be derived through a simple transformation.

Example 2.

Consider the network shown in Figure 2:

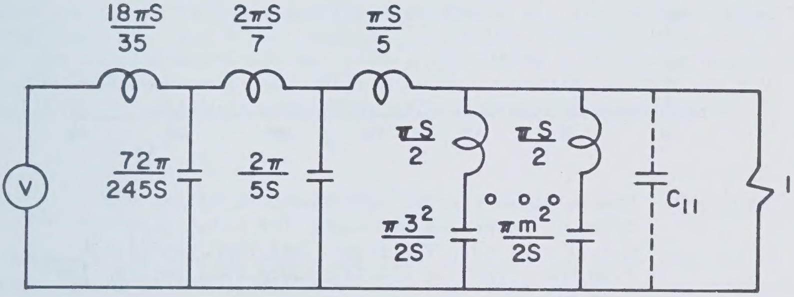


Figure 2: The F2C square pulse matched filter. This network differs from the F2 filter [5] by the added terminating capacitor C_{11}. The filter has idlers up to $m = 10$ and a terminating capacitor of $C_{11} = 10 \times C_{10}$.

This network has an impulse response confined to interval $0 \le t \le 2\pi$. The impulse response of a 4-idlers realization of this network (F2 network) has been demonstrated to perform very well as a $\sin^4 t/2$ matched time-limited filter with no negative undershoot and a small positive ripple of amplitude of 2.5% of the main signal [5]. The purpose of the present analysis is to verify the impact of deviations from the ideal design on the performance of the time limited filter. Two situations have been considered: a) the impact of the input capacity of a following amplifier on the termination of the network, and b) the impact of stray capacity. Depending on its magnitude, the added capacitor C_{11} in Figure 2 represents both situations. The F2 network considered here has four more idlers (up to $m = 10$). This improves the overall performance of the network, reducing the magnitude of the trailing ripple by a factor of two. Figure 3 demonstrates the performance of the network with a terminating capacitor 10 times larger than that of the last idler, showing no visible ripple. The figure compares the transfer functions for pulse width of 0.4, 1 and 10, illustrating the noise rejection properties of the filter. Figure 4 illustrates the sensitivity of the filter to the tolerance of the added capacitor. The modified filter is denoted as F2C , followed by a number indicating the impedance ratio between the additional capacitor and that of the $m = 10$ idler. It is demonstrated (Figure 4) that the trailing ripple is virtually eliminated, using a termination capacitative impedance of 0.104 of the impedance of the capacitor C_{10} in the $m = 10$

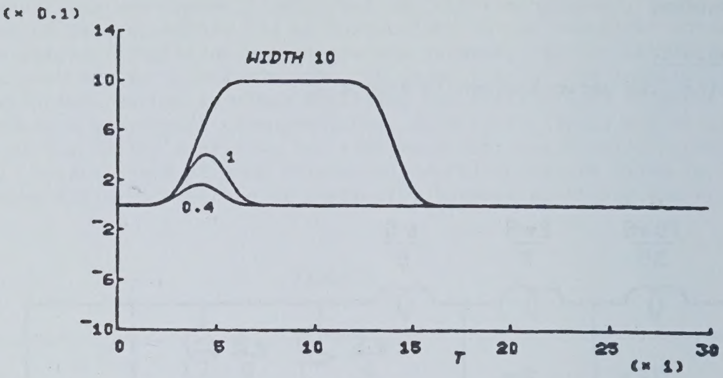


Figure 3: Square pulse transmission function of the F2C filter. Responses are shown for pulse width of 0.4, 1, and 10. The figure has been reproduced from the screen of the Tektronix 4013 Interactive Terminal.

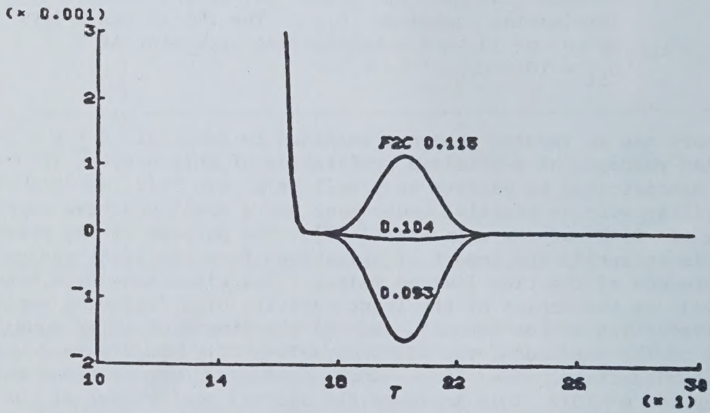


Figure 4: The trailing ripple for the response function of the F2C filter. Results are shown for C_{10}/C_{11} values of 0.093, 0.104 and 0.115, representing ±10% tolerance over the best match of 0.104. The respective magnitudes of the trailing ripple are of less than ± 0.002 of that of the input square signal. The figure has been reproduced from the screen of of the Tektronix 4013 Interactive Terminal.

idler. Also shown are the response curves for ± 10% tolerance in the terminating capacitor, demonstrating a ripple magnitude of less than ± 0.002 of the amplitude of the input signal.

Using a small termination capacitor as representative of stray capacitance is more of a test for the integration algorithm than for the robustness of the filter. Assigning to the termination capacitor a capacitative impedance 20 times that of C_{10}, gives rise to a high frequency near instability of the network, which produces a numerical instability of the explicit integration procedure. Using a 5th order polynomial time prediction, the best results for the ideal network ($C_{11} = 0$) are obtained using the Taylor's coefficients (Table I), with Maximal time interval size of $t \simeq 0.34$. Setting $C_{11} = 0.05 \times C_{10}$, reduced Δt_{max} for the A50 predictor to 0.001, indicating a stiffness ratio of 340. The Chebychev predictor (A14) provides a stable integration for $\Delta t_{max} \simeq 0.0155$, which is consistent with theoretical expectations.

DISCUSSION

Although the present paper was restricted to the discussion of strictly linear networks, the numerical algorithm described is more general and is applicable to non-linear networks as well. Its application to the initial value problem for non-linear systems of equations has been discussed elsewhere [3]. The emphasis in the present paper is on the extension of the method to the boundary value problem. The algorithm has also been applied successfully to switching networks as well as to the problem of non-linear termination of the RLC filter, where the termination resistor is voltage dependent. This is the case for a termination with a diode, or the input impedance of a following amplifier. Further computational efficiency may be realized if the analysis is restricted *a priori* to linear networks only, making implicit integration methods more attractive.

REFERENCES

1. R.W. Jensen and L.P. McNamee, *Handbook of Circuit Analysis Languages and Techniques*, Prentice Hall, Inc. (1976).

2. E.S. Kuh and R.A. Rohrer, "The State Variable Approach to Network Analysis," Proceedings IEEE, 53, July 1965, pp. 672-686.

3. Y. Hazony, "Extended Stability Range for the Explicit Integration of the Initial Value Problem," *Bull. Amer. Phys. Soc.*, 22, 7-8, January 1977).

4. P.J. Van der Houwen, *Construction of Integration Formulas for Initial Value Problems*, North-Holland Publishing Company, 1977.

5. D. Hazony and Y. Hazony, "Time Limited RLC Networks," Proceedings paper, this issue.

SINGULAR POLE ANALYSIS OF ACTIVE FILTERS

A. M. DAVIS AND E. A. MOUSTAKAS
San Jose State University, San Jose, California 95192

ABSTRACT

It is possible to analyze active networks by computing first order poles and resistive transfer functions. This paper applies the above "singular pole" analysis technique to develop general expressions for the voltage transfer functions of active networks with two and three capacitors. Some general results on sensitivities are presented, and several important active filter topologies are analyzed to illustrate the power of the method.

INTRODUCTION

The analysis of linear active or passive RC networks is easily accomplished by the "singular pole" method [1]. This consists merely of calculation of first order poles due to each capacitor with the others variously shorted or open circuited, as well as the calculation of resistive transfer functions under similar conditions. Reference [1] develops the procedure, which is a completion of ideas contained in references [2] and [3]. The thrust of this paper is to use the singular pole technique to develop general expressions for the voltage transfer functions of networks containing two and three capacitors. The form of these expressions allows classification of filter type (low pass, band pass, etc.) by inspection as well as rapid calculation of the transfer function itself.

THE THREE-CAPACITOR NETWORK

A general expression for the transfer function of a three capacitor network can be derived from bilinearity considerations [1] as

$$H(s) = \frac{H_{123}a_{123}s^3 + (H_{12}a_{12}+H_{13}a_{13}+H_{23}a_{23})s^2 + (H_1a_1+H_2a_2+H_3a_3)s + H_0}{a_{123}s^3 + (a_{12}+a_{13}+a_{23})s^2 + (a_1+a_2+a_3)s + 1} \tag{1}$$

where the H's are resistive transfer functions calculated with the capacitors corresponding to the subscripts shorted and the others open circuited.

The a's can be interpreted as products of RC time constants or inverse first order pole magnitudes. These products are formed as follows. In the computation of $a_{ijk\ldots}$, all capacitors with subscripts not belonging to the set of subscripts of $a_{ijk\ldots}$ are open circuited. Any subscript of $a_{ijk\ldots}$, say j, is then selected and the inverse pole magnitude due to c_j is calculated with the capacitors corresponding to the other subscripts of $a_{ijk\ldots}$ shorted. Capacitor c_j is then open circuited, and j is removed from the set of subscripts under consideration. Any other subscript from the reduced set is selected and the process above is repeated. This continues until all subscripts are exhausted.

For example one can write $a_{13}=(P_3^1 \cdot P_1)^{-1}$, where P_3^1 is the pole due to

c_3 with c_1 shorted and c_2 opened, and P_1 is the pole due to c_1 with c_2 and c_3 open circuited. One could also write $a_{13}=(P_1^3 \cdot P_3)^{-1}$.

THE TWO-CAPACITOR NETWORK

For a network containing two capacitors (such as the ubiquitous two capacitor biquad) it is profitable to express the transfer function in terms of pole magnitudes, rather than inverse pole magnitudes. The analysis then yields

$$H(s) = \frac{H_{12}s^2 + (H_1P_2^1+H_2P_1^2)s + H_0 \cdot \left\{ \begin{matrix} P_1^2 \cdot P_2 \\ \text{or} \\ P_2^1 \cdot P_1 \end{matrix} \right\}}{s^2 + (P_2^1+P_1^2)s + \left\{ \begin{matrix} P_1^2 \cdot P_2 \\ \text{or} \\ P_2^1 \cdot P_1 \end{matrix} \right\}} \tag{2}$$

where the H's and P's are defined as above. The choice of last term corresponds to the two different ways of calculating a two capacitor pole product. Note that if $H_1=H_2=H_{12}=0$, the network is low pass, if $H_{12}=H_0=0$ it is band pass, and so on.

If one notes that all poles above are first order, it is a simple matter to see that the resonant frequency is given by

$$\omega_0 = \text{constant} \cdot (c_1c_2)^{-1} \tag{3}$$

from which one easily derives

$$s_{c_1}^{\omega_0} = s_{c_2}^{\omega_0} = -\tfrac{1}{2} \tag{4}$$

or

$$s_{c_1}^{\omega_0} + s_{c_2}^{\omega_0} = -1 \ . \tag{4a}$$

One can also show that

$$s_{c_1}^{Q} + s_{c_2}^{Q} = 0 \, . \tag{5}$$

Results similar to (4a) and (5) hold for the three-capacitor network (and, indeed, for networks with any number of capacitors) when it yields a biquadratic transfer function by virtue of capacitor loop degeneracy.

EXAMPLES

As a first example of application of the singular pole technique, consider Figure 1 [5]. In 1(a) is shown the network to be analyzed. Clearly $H_{12}=H_1=0$ since c_1 is in shunt with the signal path. Shorting the driving voltage source results in the circuit of 1(b), from which $R_{Th}=R_3$. Thus, $P_2^1 = \frac{1}{R_{Th}c_2} = \frac{G_3}{c_2}$. 1(c) results from shorting c_2, and Miller's Theorem has been used to derive the equivalent circuit of 1(d). It is then a simple matter to see that

$$H_2 = \frac{kG_1}{G_1 + (1-k)G_2 + G_3} .$$

With $V_{in} = 0$, R_{Th} can be computed as

$$R_{Th} = \frac{1}{G_1 + (1-k)G_2 + G_3}$$

so

$$P_1^2 = \frac{G_1 + (1-k)G_2 + G_3}{c_1} .$$

Figure 1(e) results from opening both c_1 & c_2, and the topology clearly shows that $H_0 = 0$. Since $H_{12}=H_0=0$, it is evident that the circuit is band pass. Shorting the driving source, again, results in $R_{Th} = \frac{1}{G_1 + G_2}$, so $P_1 = \frac{G_1 + G_2}{c_1}$. Thus,

$$H(s) = \frac{\dfrac{kG_1}{G_1 + (1-k)G_2 + G_3} \cdot \dfrac{G_1 + (1-k)G_2 + G_3}{c_1} s}{s^2 + \left[\dfrac{G_1 + (1-k)G_2 + G_3}{c_1} + \dfrac{G_3}{c_2}\right] s + \dfrac{G_3}{c_2} \cdot \dfrac{G_1 + G_2}{c_1}}$$

or, more simply,

$$H(s) = \frac{kG_1}{c_1} \cdot \frac{s}{s^2 + \left[\dfrac{G_1 + (1-k)G_2 + G_3}{c_1} + \dfrac{G_3}{c_2}\right] s + \dfrac{G_3(G_1 + G_2)}{c_1 c_2}} .$$

The second example, shown in Figure 2 [4], is a three capacitor network. Due to space considerations, each equivalent circuit will not be shown. They are, however, simple in character and--in most cases--can be analyzed mentally. For instance, it is evident that $H_{123}=H_{12}=0$ (if $k \neq 1$) since the input and output of the gain block must be identical. $H_{13} = 0$, also, because opening c_2 forces the input to the amplifier to zero. It is straightforward to show that

$$H_{23} = \frac{kG_1}{G_1 + G_2} .$$

Judicious application of Miller's Theorem indicates that

$$H_2 = k \cdot \frac{\dfrac{R_2 + R_3}{1-k}}{R_1 + \dfrac{R_2 + R_3}{1-k}} = \frac{kG_1(G_2 + G_3)}{(1-k)G_2G_3 + G_1G_3 + G_2G_3}$$

while $H_1=H_3=H_0=0$ due to the open circuit created by c_2.

The inverse pole products are just as easily derived. For instance

$$a_{123} = \frac{c_1(1-k)}{G_1 + G_2} \cdot \frac{c_2(G_1 + G_2)}{G_1G_2} \cdot \frac{c_3}{(1-k)G_3} = \frac{c_1c_2c_3}{G_1G_2G_3} \ .$$

Also

$$a_{12} = \frac{(1-k)(G_2 + G_3)c_1}{G_1G_2 + G_1G_3 + (1-k)G_2G_3} \cdot \frac{(G_1G_2 + G_1G_3 + (1-k)G_2G_3)c_2}{(1-k)G_1G_2G_3}$$

$$= \frac{(G_2 + G_3)c_1c_2}{G_1G_2G_3}$$

$$a_{13} = \frac{c_1}{G_1} \cdot \frac{c_3}{(1-k)G_3} = \frac{c_1c_3}{(1-k)G_1G_3}$$

$$a_{23} = \frac{(G_1 + G_2)c_2}{G_1G_2} \cdot \frac{c_3}{(1-k)G_3} = \frac{(G_1 + G_2)c_2c_3}{(1-k)G_1G_2G_3}$$

$$a_1 = \frac{c_1}{G_1}$$

$$a_2 = \frac{(G_1G_2 + G_1G_3 + (1-k)G_2G_3)c_2}{(1-k)G_1G_2G_3}$$

$$a_3 = \frac{c_3}{(1-k)G_3} \ .$$

With the above values,

$$H(s) = \frac{\dfrac{kG_1}{G_1 + G_2} \cdot \dfrac{(G_1 + G_2)c_2c_3}{(1-k)G_1G_2G_3} s^2 + \dfrac{kG_1(G_2 + G_3)}{G_1G_2 + G_1G_3 + (1-k)G_2G_3} \cdot \dfrac{c_2(G_1G_2 + G_1G_3 + (1-k)G_2G_3)}{(1-k)G_1G_2G_3} s}{\dfrac{c_1c_2c_3}{G_1G_2G_3} s^3 + \left[\dfrac{(G_2 + G_3)c_1c_2}{G_1G_2G_3} + \dfrac{c_1c_3}{(1-k)G_1G_3} + \dfrac{(G_1 + G_2)c_2c_3}{(1-k)G_1G_2G_3}\right] s^2 + \left[\dfrac{c_1}{G_1} + \dfrac{(G_1G_2 + G_1G_3 + (1-k)G_2G_3)c_2}{(1-k)G_1G_2G_3} + \dfrac{c_3}{(1-k)G_3}\right] s + 1} \ .$$

Simplification yields

$$H(s) = \frac{\dfrac{kG_1}{(1-k)c_1c_3}\left(s^2 + (G_2 + G_3)s\right)}{s^3 + \left[\dfrac{G_2 + G_3}{c_3} + \dfrac{G_2}{(1-k)c_2} + \dfrac{G_1 + G_2}{(1-k)c_1}\right] s^2 + \left[\dfrac{G_2G_3}{c_2c_3} + \dfrac{G_1G_2 + G_1G_3 + (1-k)G_2G_3}{(1-k)c_1c_3} + \dfrac{G_1G_2}{(1-k)c_1c_2}\right] s + \dfrac{G_1G_2G_3}{c_1c_2c_3}}$$

which checks with reference [5]. Although the expression is somewhat complex, the procedure in deriving it is very methodical, more intuitive, and less susceptible to error than loop or nodal analysis.

The last example consists of the Friend-Deliyannis biquad [6]. It is sketched in Figure 3(a). A transformed circuit is shown in Figure 3(b), where $k = \frac{A}{1 - cA}$ with $c = \frac{R_a}{R_a + R_b}$. Miller's Theorem has been invoked to return R_2 to ground rather than V_{out}. The resulting equivalent is greatly simplified. Again, it is easy to see that : (with $k \neq 1$)

$$H_{12} = H_2 = 0$$

$$H_1 = \frac{-k \cdot \frac{R_2}{1+k}}{R_1 + \frac{R_2}{1+k}} = \frac{-kR_2}{(1+k)R_1 + R_2}$$

$H_0 = 0$ (these results indicate the circuit is band pass)

$$P_1^2 = \frac{1}{R_2 c_1}$$

$$P_2^1 = \frac{(1+k)R_1 + R_2}{(1+k)R_1 R_2 c_2}$$

$$P_1 = \frac{1+k}{((1+k)R_1 + R_2)c_1}$$

Thus,

$$H(s) = \frac{\frac{-kR_2}{(1+k)R_1 + R_2} \cdot \frac{(1+k)R_1 + R_2}{(1+k)R_1 R_2 c_2} s}{s^2 + \left[\frac{1}{R_2 c_1} + \frac{(1+k)R_1 + R_2}{(1+k)R_1 R_2 c_2}\right] s + \frac{(1+k)R_1 + R_2}{(1+k)R_1 R_2 c_2} \cdot \frac{1+k}{((1+k)R_1 + R_2)c_1}}$$

or

$$H(s) = \frac{\frac{-k}{1+k} \cdot \frac{1}{R_1 c_2} s}{s^2 + \left[\frac{1}{R_2 c_1} + \frac{(1+k)R_1 + R_2}{(1+k)R_1 R_2 c_2}\right] s + \frac{1}{R_1 R_2 c_1 c_2}}$$

which checks with the analysis presented in [6].

CONCLUSION

The singular pole analysis technique permits rapid analysis of active (or passive) RC networks, as well as classification of filter type, usually by inspection. The power of the approach is exemplified in the examples treated.

REFERENCES

1. A. M. Davis and E. A. Moustakas, "Singular Pole Analysis of Circuits," submitted to IEEE Transactions on Circuits and Systems.

2. R. D. Thornton, et al., Multistage Transistor Circuits, John Wiley, 1965, Ch. 1.

3. B. L. Cochrun and A. Grabel, "A Method For the Determination of the Transfer Function of Electronic Circuits," IEEE Transactions on Circuits and Systems, Vol. CT-20, No. 1, January, 1973.

4. M. A. Soderstrand and S. K. Mitra, "Sensitivity Analysis of Third Order Filters," International Journal of Electronics, Vol. 30, No. 3, pp. 265-272.

5. A. Budak, Passive and Active Network Analysis and Synthesis, Houghton Mifflin, 1974.

6. P. E. Fleisher, "Sensitivity Minimization in a Single Amplifier Biquad Circuit," IEEE Transactions on Circuits and Systems, CAS-23, No. 1, January, 1976.

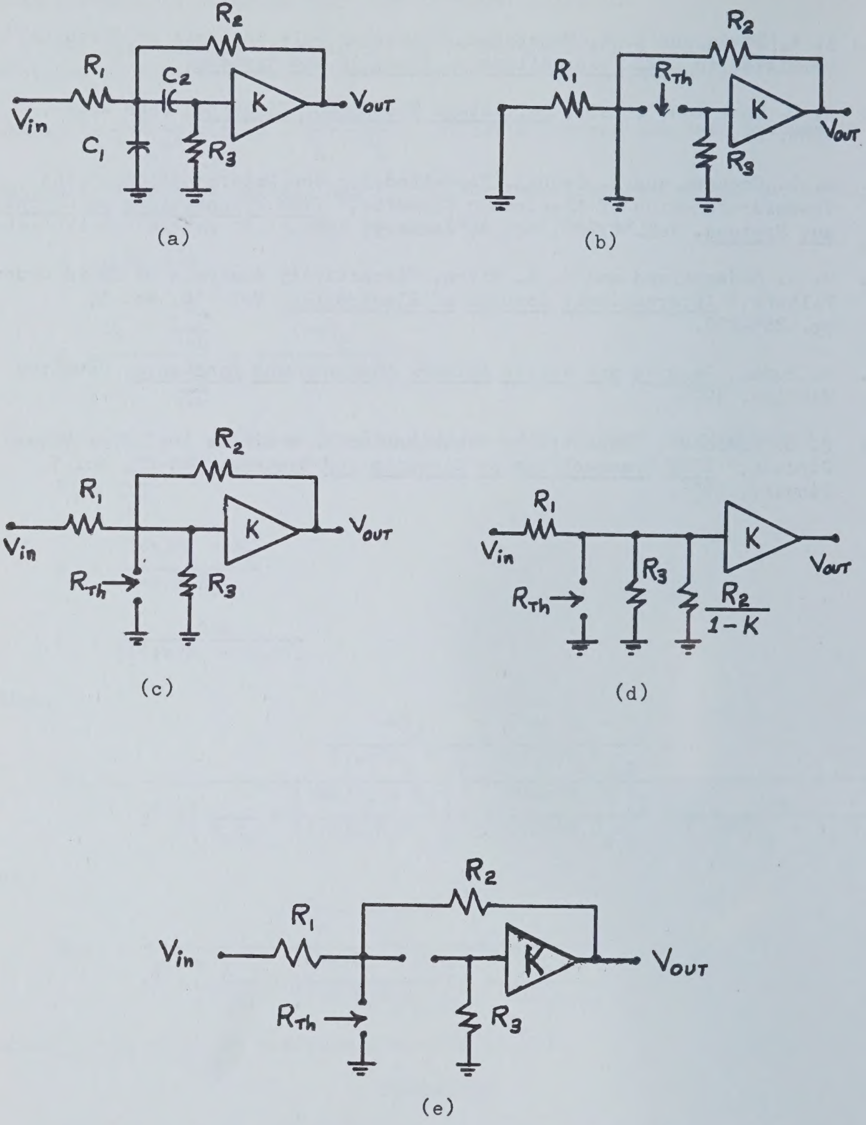

R2
R1
C2
Vin
K
VOUT
C1
R3
(a)
R2
R1
RTh
K
VOUT
R3
(b)
R2
R1
Vin
K
VOUT
RTh
R3
(c)
R1
Vin
K
VOUT
RTh
R3
R2/(1-K)
(d)
R2
R1
Vin
K
VOUT
RTh
R3
(e)

FIGURE 1

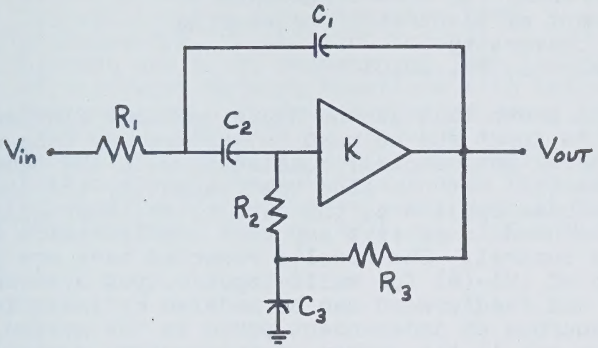

C₁
R₁
C₂
Vin
K
VOUT
R₂
R₃
C₃

FIGURE 2

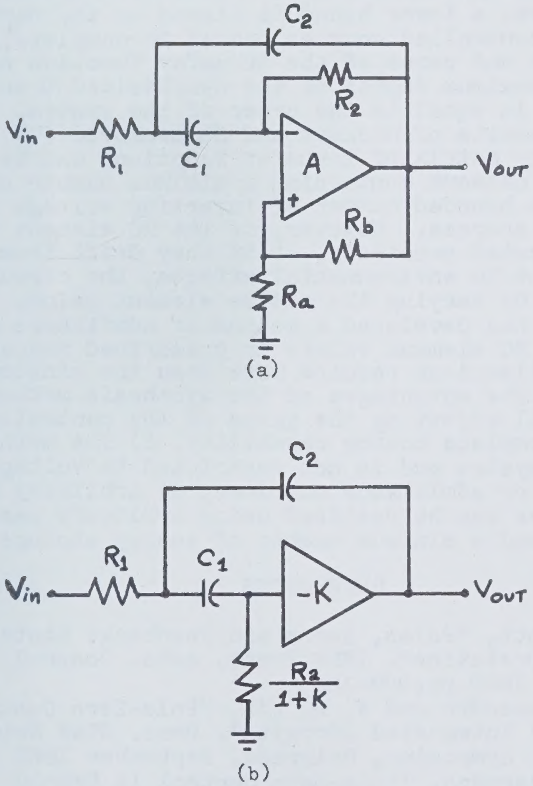

C₂
R₂
Vin
R₁
C₁
−
A
VOUT
+
Rb
Ra
(a)
C₂
R₁
C₁
Vin
−K
VOUT
R₂/(1+K)
(b)

FIGURE 3

MODIFICATION OF LUMPED LINEAR TIME-INVARIANT SYSTEMS BY CONTROLLED SOURCES FOR TRANSFER FUNCTION MATRIX SYNTHESIS

RICHARD PERRY and JOHN D'ALESSANDRO
Department of Electrical Engineering
Drexel University
Philadelphia, Pa. 19104

It is well known that linear state variable feedback applied to the input can be used to achieve any pole configuration (natural frequencies) consistent with the dimensions of the system (1). It has also been shown (2)-(6) that state variable feedback applied to the output, or internally to a chord, can be used to achieve any zero configuration independent of pole control. The results reported here are primarily an extension of (2)-(6) for multi-input/output systems.

Feedback and feedforward can be modeled by inserting controlled sources at independent ports in the system. By a consideration of: 1) the number of independent variables available for use as inputs to a multiple-input controlled source and 2) the minimal number of parameters needed to define the system, a lower bound is placed on the number of multiple-input controlled sources needed to completely control all of the poles and zeros of the transfer function matrix. Of course, the maximum degree of the synthesized transfer function matrix is equal to the order of the system.

The recent results of Bickart and Stamenković (7) show that an arbitrary matrix of transfer functions can be realized by an active RC network containing a minimum number of capacitors and a bounded number of inverting voltage controlled voltage sources. However, if the RC element values cannot be fabricated precisely, or if they drift from their design values due to environmental effects, the circuit can not be re-tuned by varying the active element gains. Stamenković (8) has developed a method of admittance matrix synthesis using RC element values in prescribed ranges, however the realizations require more than the minimum number of capacitors. The advantages of the synthesis method I am proposing are: 1) Adjusting the gains of the controlled sources gives complete tuning capability, 2) The method applies to any system and is not restricted to voltage transfer ratios or admittance matrices, 3) Arbitrary transfer function matrices can be realized using arbitrary passive element values and a minimum number of energy storage elements.

REFERENCES

(1) R. W. Brockett, "Poles, Zeros and Feedback: State Space Interpretation". IEEE Trans. Auto. Control April 1965 pp.129-135

(2) J. R. D'Alessandro and W. H. Kim, "Pole-Zero Control in Hybrid Integrated Circuits". Proc. ETAN Network Theory Symposium, Belgrade, September 1968

(3) J. R. D'Alessandro, "Pole-Zero Control in Hybrid Integrated Circuits". IEEE Int. Symposium on Circuit Theory, San Francisco, Ca. December 1969

(4) J. R. D'Alessandro, U. S. Patent #3,643,184 Feb. 1972, "Multiport Feedback and Pole-Zero Control".
(5) J. R. D'Alessandro, "The Ratio Design Problem, Pole-Zero Control, and Applications". Thesis, Columbia University, New York, November 1968.
(6) J. R. D'Alessandro, Unpublished Works.
(7) T. A. Bickart and B. B. Stamenković, "On the Realization of Multiport Network Functions with Active RC Networks and a Minimum Number of Capacitors". IEEE Proc. ISCAS/76 pp. 287-290
(8) B. B. Stamenković, "On the Active RC Realization of Rational Admittance Matrices with Prescribed Ranges of Passive RC Element Values". Circuit Theory and Applications, Vol. 3, 1975, pp. 157-175

TIME LIMITED RLC NETWORKS

D. HAZONY
Electrical Engineering and Applied Physics Department
Case Western Reserve University, Cleveland, Ohio

Y. HAZONY
School of Engineering and Applied Science
Princeton University, Princeton, New Jersey

ABSTRACT

A great deal of present technology is governed by pulses. As the pulses travel through the system they change their characteristics and may have to undergo some filtering. Conventional filters tend to distort the signal, expand it, and reduce its symmetry. Thus a synthesis procedure is proposed that produces time limited RLC filters approximations to an arbitrary degree of accuracy. Many examples are shown.

RLC APPROXIMATIONS OF TIME LIMITED FILTERS

1. Introduction

Of concern are problems of engineering and physics where the principle signals are pulses. Our technological community has been exposed of late to a great deal of activity pertaining to sharp pulses. Alas, but as pulses travel through the system they are corrupted by noise, infested with ripples and often undergo severe distortions. Thus it is proposed to design special filter elements capable of interacting with the signal in a general way but in a time limited fashion.

Time limited devices may represent many of the conventional filters like low-pass, band-pass, high-pass, and matched filters with the additional constraint that the respective impulse response, h(t), is governed by:

$$\begin{aligned} h(t) &\not\equiv 0 \qquad 0 < t < T \\ &= 0 \qquad \text{elsewhere} \end{aligned} \tag{1}$$

Many of the surface acoustic wave devices as well as those that employ delay lines are time limited. Similarly, it is easy to devise digital filters that are time limited. The purpose of this work is to realize RLC filters which are, for all practical purposes, time limited. This means that outside of the active region 0,T, spurious responses can be made arbitrarily small.

The fact that h(t) is time limited, restricts the LaPlace Transform H(s) (= L(h)) to have no poles in the entire s plane. Hence, H(s) is said to be entire. Diamond and Gerst discuss these functions extensively in reference 1.

For example, a square pulse of 2π second duration, is Laplace Transformed to give:

$$H(s) = \frac{1-e^{-2\pi s}}{s} \tag{2}$$

A study of this function near the origin shows that there is no pole there. Hence, H(s) is entire. Similarly, the following is also entire.

$$H(s) = \left(\frac{1}{s} + \frac{2s}{s^2+1}\right)\left(1-e^{-2\pi s}\right). \tag{3}$$

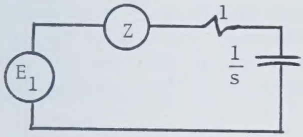
Functions of this kind will be used throughout the paper.

2. Development

Suppose that a device is made whose driving point impedance function Z is given by

$$Z = \coth s - 1/s \ . \tag{4}$$

Similar functions are obtained in the study of tapered transmission lines. R. W. Newcomb gives an example of this kind in reference 2. Also, see reference 3. Using this device in the following circuit (Figure 1) gives:



$$\frac{E_2}{E_1} = \frac{\frac{1}{s}}{(\coth s - \frac{1}{s}) + \frac{1}{s} + 1} \tag{5}$$

$$= \frac{\sinh s}{s} e^{-s}$$

Fig.1 - Synthesis of $e^{-s}\sinh s/s$

Thus the negative element in Equation 4 is used to eliminate the 1/s term in the denominator, producing an entire function. It follows that more advantages would be derived if Z in Equation 4 had additional poles missing.

Continuing the development we use the admittance form:

$$Y = \coth \pi s - Y_{22} \ . \tag{6}$$

This element is now used in Figure 2.

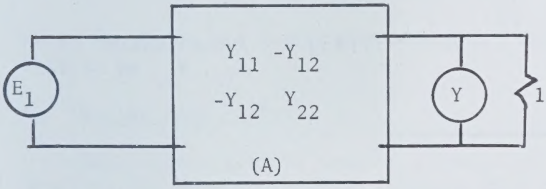


Fig.2 - The Synthesis Prototype

The voltage transfer function of a two-port network is given by:

$$t_{21} = -Y_{12}/Y_{22} \tag{7}$$

It follows for the network shown in Figure 2 that t_{21} is given by:

$$t_{21} = \frac{-Y_{12}}{Y_{22} + Y + 1}$$

$$= \frac{Y_{12}}{Y_{22} + \coth \pi s - Y_{22} + 1} \tag{8}$$

$$= - e^{-\pi s} Y_{12} \sinh \pi s = - \frac{1}{2} Y_{12}(1-e^{-2\pi s}) \ .$$

A parallel development employing tanhπs instead of cothπs yields

$$t_{21} = -\frac{1}{2} Y_{12}(1+e^{-2\pi s}) = -Y_{12}e^{-\pi s} \cosh \pi s. \tag{9}$$

Example 1: Construct a filter whose impulse response is given by:

$$\begin{aligned} h(t) &= \frac{1}{\pi} \sin^2 t/2 && 0 \leq t \leq 2\pi \\ &= 0 && \text{elsewhere} . \end{aligned} \tag{10}$$

Solution: $L(h(t))$ is given by

$$H(s) = \frac{1}{2\pi s(s^2+1)}(1-e^{-2\pi s}) = \frac{1}{\pi s(s^2+1)} \sinh \pi s \; e^{-\pi s} . \tag{11}$$

It follows from Equations 6, 8, and 11, that:

$$-Y_{12} = \frac{1}{\pi s(s^2+1)} \tag{12}$$

$$Y_{22} = \frac{1}{\pi}\left(\frac{1}{s} + \frac{2s}{s^2+1}\right) \rightarrow \text{(arbitrary at this point)}$$

$$Y = \coth \pi s - Y_{22} .$$

It is seen that the synthesis of $H(s)$ was transformed into the synthesis of a two port network where $-Y_{12}$ and Y_{22} are prescribed. Necessary and sufficient conditions for realizing such a network are discussed by Fialkow and Gerst[4]. A ladder network realization is shown in Figure 3.

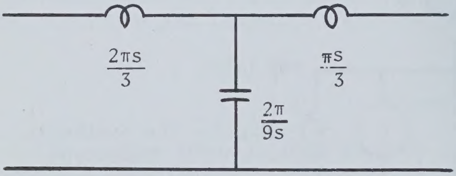


Fig.3 - A Realization of Y_{12}, Y_{22} in Eq.12.

This network represents section A in Figure 2.

Example 2: Realize:

$$\begin{aligned} h(t) &= \frac{1}{\pi} \sin^2 t && 0 < t < 2\pi \\ &= 0 && \text{elsewhere} \end{aligned}$$

Solution:

$$H(s) = \frac{4}{\pi s(s^2+4)} \sinh \pi s \; e^{-\pi s}$$

$$-Y_{12} = \frac{4}{\pi s(s^2+4)} \qquad \text{(by Eq. 8)}$$

$$Y_{22} = \frac{1}{\pi}\left(\frac{1}{s} + \frac{2s}{s^2+4}\right) \text{ (for realization ease).}$$

The network (A) in Figure 2 will be the same as in Figure 3, except that the capacitor will be given by $\frac{8\pi}{9s}$.

3. The Synthesis

Repeating Equations 8 and 9,

$$t_{21} = \frac{1}{2} Y_{12} (1 - e^{-2\pi s}) \tag{8}$$

or

$$t_{21} = -\frac{1}{2} Y_{12} (1 + e^{-2\pi s}) . \tag{9}$$

It is seen that the voltage transfer function synthesis is reduced to that of $-Y_{12}$ of some 2-port network. In addition, if the impulse h(t) is time limited, t_{21} must be entire. This means that the poles of Y_{12} must be zeros of either $1-e^{-2\pi s}$ or $1+e^{-2\pi s}$. Hence

$$-Y_{12} = \frac{a_n s}{s^2 + n^2} \qquad \text{(associating with Eq. 8)} \tag{13}$$

$$-Y_{12} = \frac{b_n s}{s^2 + (\frac{n}{2})^2} \qquad \text{(associating with Eq. 9)} \quad (n = \text{odd}) \tag{14}$$

Moreover, according to Fialkow and Gerst[4], Y_{22} must also be of the same form except that the respective coefficients must be positive. Of course, the procedure is predicated on the existence of elements whose driving point admittance functions are given by:

$$Y = \coth \pi s - Y_{22}$$

or

$$Y = \tanh \pi s - Y_{22}$$

These functions are easily approximated. This will be discussed in Section 5.

4. The Low-Pass Filter

This filter will be used as a practical example for the procedure. Usually, the impulse response of a low-pass filter is not symmetric. Consequently the symmetry of a processed signal is reduced. This can be avoided. The filter will be described in terms of a family, $F_n(s)$: (see ref. 5)

$$F_o = \frac{e^{-\pi s} \sinh \pi s}{s\pi} \tag{15}$$

$$F_1 = F_o(1+s^2)^{-1}$$

$$F_2 = F_1(1+s^2/4)^{-1}$$

$$F_3 = F_2(1+s^2/9)^{-1}$$

. . . .

$$F_n = F_{n-1}(1+s^2/n^2)^{-1}$$

The impulse response of F_o is a square pulse, that of F_1 is a squared sine as in Equation 10, and in general,

$$h_n(t) = \frac{\sin^{2n} t/2}{\pi(2n)!} (2^n n!)^2 \qquad 0 < t < 2\pi$$
$$= 0 \qquad \text{elsewhere}$$

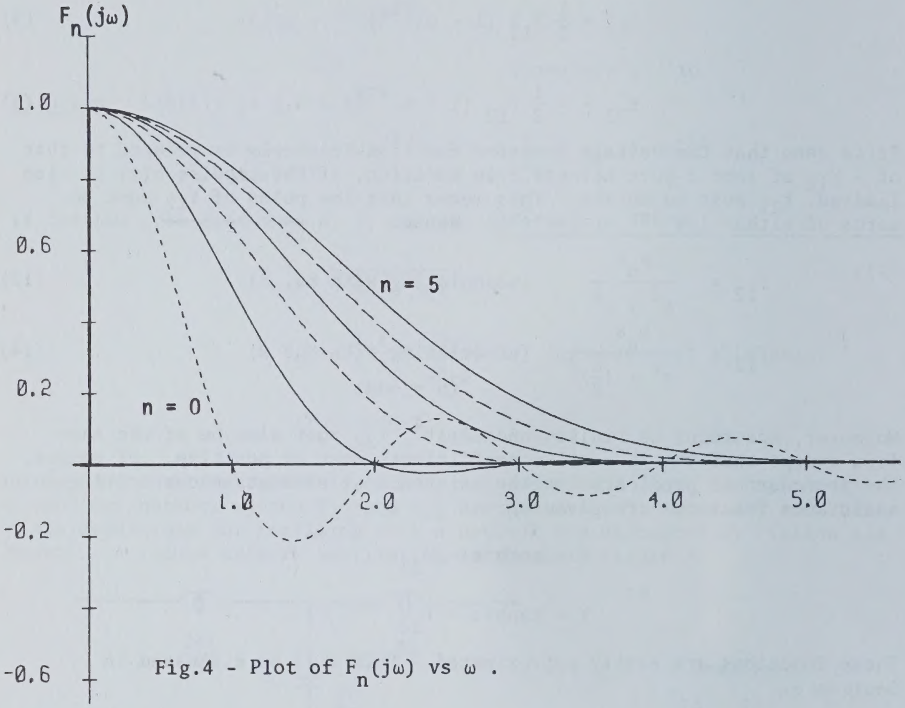


Fig.4 - Plot of $F_n(j\omega)$ vs ω .

The gain of $F_n(j\omega)$ is shown in Figure 4. Realizations of the 2-port (A) of Figure 2 are shown below:

F_o $$-Y_{12} = Y_{22} = \frac{1}{\pi s}$$

F_1 $$-Y_{12} = \frac{1}{\pi s(s^2+1)}$$
$$Y_{22} = \frac{1}{\pi}\left(\frac{1}{s} + \frac{2s}{s^2+1}\right)$$

F_2 $$-Y_{12} = \frac{4}{\pi s(s^2+1)(s^2+4)}$$
$$Y_{22} = \frac{1}{\pi}\left(\frac{1}{s} + \frac{2s}{s^2+1} + \frac{2s}{s^2+4}\right)$$

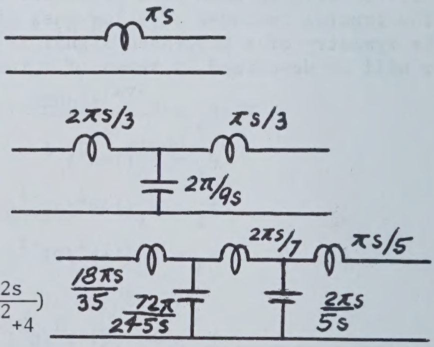


Figure 5 - Realizations of (A) in Figure 2 for F_o, F_1, and F_2 . The general form will be a ladder network.

It will be shown in the next section that these filters act like thick-pass filters. Thick pulses are passed with their symmetry intact. Only the corners are rounded. Thin pulses are severely attenuated. (See Fig. 7).

5. The Approximation Problem

The key element in the synthesis so far has been the ability to generate transmission line admittance functions with poles missing. This problem is minimized if the hyperbolic tangent and cotangent are first approximated by rational admittance functions. The solution is not unique. Let these functions be expanded in partial fractions:

$$Y_{na} = \coth\pi s \doteq \frac{1}{\pi}\left(\frac{1}{s} + \frac{2s}{s^2+1} + \frac{2s}{s^2+4} + \frac{2s}{s^2+9} + \ldots + \frac{2s}{s^2+n^2}\right) \tag{17}$$

$$Y_{nb} = \tanh\pi s \doteq \frac{2}{\pi}\left(\frac{s}{s^2+1/4} + \frac{s}{s^2+9/4} + \frac{s}{s^2+25/4} + \frac{s}{s^2+49/4} + \ldots + \frac{s}{s^2(2n+1)^2/4}\right)$$

Here, n represents the number of terms in the approximation. It follows that Y_{22} is obtained by selecting any number of these poles (or parts of them). In the earlier examples Y_{22} was selected to give a minimum number of elements.

Example 3: Use Y_{na} in Eq. 17 to complete the networks F_o, F_1, and F_2 (Figure 5).

Solution: Take early terms of Y_{na} to form Y_{22}. The rest of the terms (poles) are realized as idlers in parallel with the load. The results are shown in Figure 6.

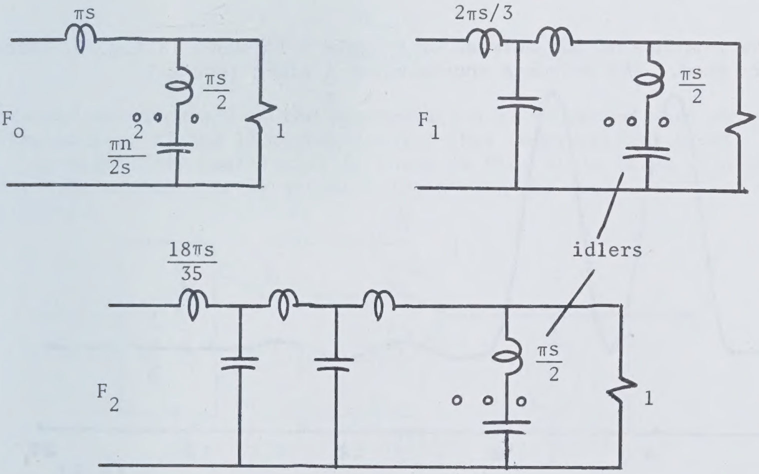


Fig.6 - Realization of F_o, F_1, and F_2 (Eq. 15)

The response of F_2 to a sequence of squares is shown in Figure 7 below. Note the symmetry of the response and the weakness of errors* in the forbidden zones.

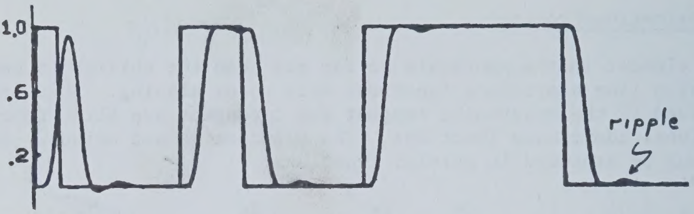


Fig.7 - F_2 (n=5) Response to Various Square Pulses

The impulse response of F_o is shown in Fig. 8.

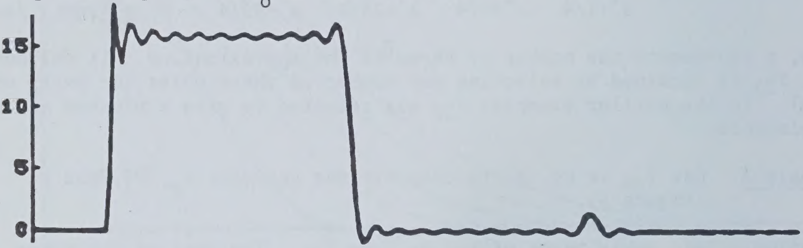


Fig.8 - F_o Impulse Response for n = 10.

The impulse response of the network of Example 2 is shown in Fig. 9. Nine idlers were used. The response approximates $\frac{1}{\pi}\sin^2 t$ ($0<t<2\pi$).

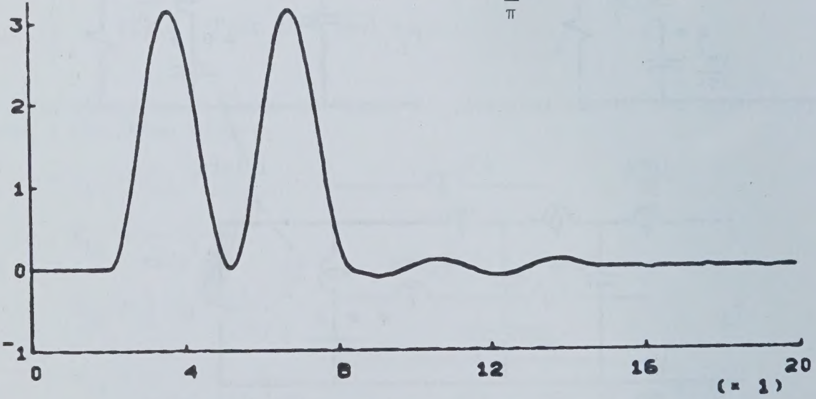


Fig.9 - Example 2 Impulse Response (9 - idlers).

---

*It is shown in an accompanying paper that this ripple is made to virtually vanish by adding an appropriate capacitor in parallel with the load [6].

6. Synthesis of Arbitrary Shapes

The main approach will be developed through an example. It relies on Fourier Series Expansion. Suppose it is necessary to design a network whose impulse response is a doublet as shown in Fig. 10.

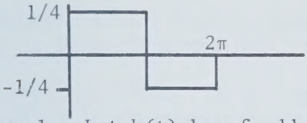


Fig.10 - A Square Doublet

Theorem 1. Let h(t) be of odd symmetry and confined to the interval $0 < t < 2\pi$. Let the Fourier series of h(t) and its transform be given by:

$$h(t) = \frac{1}{\pi}\sum_{0}^{n} b_n \cos(n + \tfrac{1}{2})t \qquad (18)$$

$$= \frac{1}{\pi}\left(\cos t/2 - \frac{1}{3}\cos 3t/2 + \frac{1}{5}\cos 5t/2 - \frac{1}{7}\cos 7t/2\right)$$

$$H(s) = \frac{1}{\pi}\left(\frac{s}{s^{1}+1/4} - \frac{s/3}{s^{2}+9/4} + \frac{s/5}{s^{2}+25/4} - \frac{s/7}{s^{2}+49/4}\right)(1+e^{-2\pi s}) \qquad (19)$$

$$= 2e^{-\pi s}H_2(s)\cosh\pi s$$

Then it is possible to realize a network where $-Y_{12} = 2H_2(s)$ and $Y_{22} \simeq \tan\pi s$ (Eq. 17) terminated in a 1 ohm whose impulse response approximates h(t) to an arbitrary degree of accuracy.

Proof: The voltage transfer function of the network is:

$$-\frac{Y_{12}}{Y_{22}} = \frac{2H_2(s)}{\tanh\pi s+1} = 2e^{-\pi s}H_2\ (s)\cosh s \qquad (20)$$

Moreover the accuracy of the approximation is increased when more terms are taken in Eqs. 17 and 18 respectively. This completes the proof.

A simplified realization is shown in Fig. 11 in terms of a phase splitter. The negative terms are associated with the negative output of the splitter.

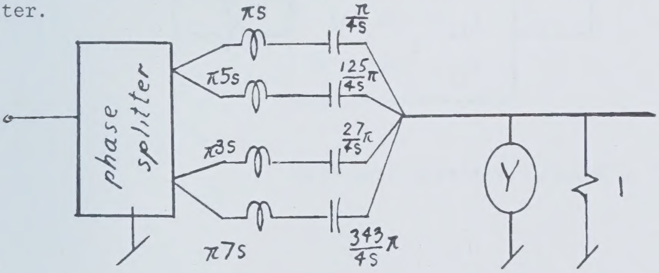


Fig.11 - 4-Term Realization of Fig.10.

$$\text{Where } Y = \tanh\pi s - \frac{2s/\pi}{s^{2}+1/4} - \frac{2s/3\pi}{s^{2}+9/4} - \frac{4/5\pi}{s^{2}+25/4} - \frac{2s/7\pi}{s^{2}+49/4}$$

Similarly we may prove the following theorems:

Theorem 2. Let h(t) be of even symmetry and confined to the interval $0 < t << 2\pi$. Let the Fourier series of h(t) and its transform be given by:

$$h(t) = \frac{1}{\pi} \sum_{o}^{n} a_n \cos nt$$

$$H(s) = \frac{1}{\pi} \sum_{o}^{n} \frac{a_n s}{s^2+n^2} (1-e^{-2\pi s}) \doteq 2e^{-\pi s} H_1(s) \sinh \pi s$$

Then it is possible to realize a network where $-Y_{12} = 2H_1(s)$ and $Y_{22} \simeq \coth \pi s$ (Eq. 17) terminated in a 1 ohm resistor whose impulse response approximates h(t) to an arbitrary degree of accuracy.

Theorem 3. Let h(t) be confined to the interval $0 < t < 2\pi$. Let the Fourier series of h(t) and its transform be given by:

$$h(t) = \frac{1}{\pi} \sum_{o}^{n} a_n \cos nt + \frac{1}{\pi} \sum_{o}^{n} b_n \cos(n + \frac{1}{2})t$$

$$H(s) = 2e^{-\pi s} (H_1(s) \sinh \pi s + H_2(s) \cosh \pi s)$$

Then the following network (Fig.12) approximates h(t) to an arbitrary degree of accuracy.

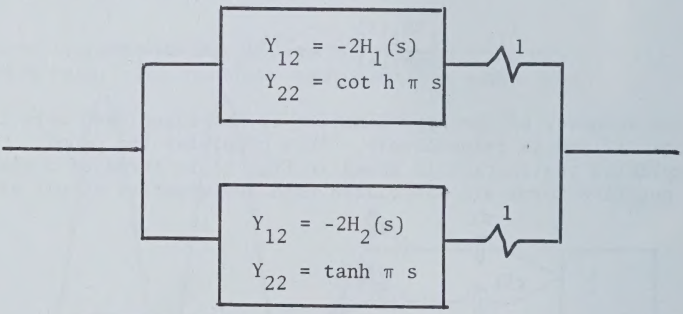


Fig.12 - A General Synthesis Prototype

7. Discussion

A synthesis is presented for networks with a prescribed impulse response. The synthesis is based on the use of incomplete transmission lines. The missing part of the transmission line acts like a negative impedance source which can be utilized to an advantage. It is shown by means of examples that the method can be utilized to synthesize conventional filters with time limited features.

REFERENCES

1. Gerst, I., and Diamond, J., "The Elimination of Intersymbol Interference by Input Signal Shaping", Proc. IRE, vol.49, pp.1195-1203, 1961.

2. Newcomb, R. W., "Active Integrated Circuits Synthesis", Chapter 1., Eq. 1.6.3a, Prentice Hall, Englewood Cliffs, N.J.

3. Burke, V., Duffin, R. J., and Hazony, D., "Distortionless Wave Propagation in Inhomogeneous Media and Transmission Lines", Quarterly of Applied Mathematics, pp.183-194, July 1976.

4. Fialkow, A. D., and Gerst, I., "The Transfer Function of Networks Without Mutual Reactance", Quart. Appl. Math., Vol.12, pp. 117-131, 1954.

5. Hazony, D., "Ideal Pulse Compression via Distributed Feedback", Circuit Th. and Applications, Vol.5, pp.239-243, 1977.

6. Hazony, Y., and Hazony, D., "By-Passing the Frequency Domain", An earlier paper in this session.

STABILITY RESULTS FOR A CLASS OF SYSTEMS WITH MULTIPLICATIVE STATE NOISE

GARY L. WISE and STEVEN I. MARCUS
Department of Electrical Engineering
University of Texas at Austin
Austin, Texas 78712

ABSTRACT

The stochastic stability of linear systems with non-Gaussian multiplicative (state dependent) noise is analyzed. The particular noise processes considered are a form of filtered Poisson processes. A technique is presented for investigating the p^{th} moment asymptotic stability of linear systems satisfying certain Lie-algebraic conditions. Several examples are given to illustrate the technique.

INTRODUCTION

The analysis of linear systems with multiplicative or state dependent noise (i.e. bilinear stochastic systems) has recently attracted a great deal of attention [1-14]. In particular, the stochastic stability of such systems for which the noise processes are Gaussian has been studied in some detail. In this paper the stochastic stability of a class of linear systems with non-Gaussian multiplicative noise is analyzed. The particular noise processes considered are a form of filtered Poisson processes.

Our technique for investigating stochastic stability is based upon a method used by Willsky, Marcus, and Martin [1]. In the following sections we describe a technique for investigating the stochastic stability of a class of systems. The technique is illustrated with examples.

PRELIMINARIES

We are interested in systems of the form

$$\dot{x}(t) = \left[A_0 + \sum_{i=1}^{k} A_i f_i(t) \right] x(t) \quad , \tag{1}$$

where the A_i are known n×n matrices, $x(t)$ is an n dimensional vector, and $f_1(t), \ldots , f_k(t)$ are noise processes. We assume that the noise processes are independent of the initial condition $x(0)$.

A Lie algebra of n×n matrices is a subspace of n×n matrices that is closed under the commutator product $[A,B] \triangleq AB - BA$. The Lie algebra generated by a set of n×n matrices is the smallest Lie algebra containing that set of matrices. A Lie algebra is said to be solvable if and only if there exists a nonsingular matrix P (possibly complex valued) such that PAP^{-1} is upper triangular for every A belonging to the Lie algebra (see [1]). Let $\mathscr{L}$ denote the Lie algebra generated by $A_0, A_1, \ldots , A_k$ and assume that $\mathscr{L}$ is solvable.

Let p be a positive integer and let $x_i(t)$ denote the i^{th} component of the vector $x(t)$. We say that the null solution of (1) is p^{th} moment asymptotically stable if, for all initial states $x(0)$ possessing the desired moments and independent of $f_1(t), \ldots , f_k(t)$, we have

$$\lim_{t\to\infty} E\left\{ \left[x_1(t)\right]^{p_1} \left[x_2(t)\right]^{p_2} \cdots \left[x_n(t)\right]^{p_n} \right\} = 0$$

for any set of nonnegative integers p_i that sum to p. It is shown in [1] that, if $\mathscr{L}$ is solvable, then the p^{th} moment asymptotic stability of (1) is equivalent to the first moment asymptotic stability of another system of the form (1), also with a solvable Lie algebra. Therefore, we can confine our discussion to the first moment asymptotic stability of (1), that is

$$\lim_{t\to\infty} E\left\{x_i(t)\right\} = 0 \quad , \qquad i = 1, \ldots , n .$$

DEVELOPMENT

We write the solution to (1) in the form

$$x(t) = \Phi_f(t,0)\ x(0) \quad ,$$

where $\Phi_f(t,0)$ is the transition matrix for (1), thought of as an explicit function of the processes $f_i(t)$. Since $\mathscr{L}$ is assumed to be solvable, we know that there is a nonsingular matrix P such that

$$B_i = PA_iP^{-1}$$

is upper triangular for $i = 0, 1, \ldots , k$. The equation

$$\dot{Y}(t) = \left[B_0 + \sum_{i=1}^{k} B_i f_i(t)\right] Y(t) \quad , \qquad Y(0) = I \ ,$$

where Y(t) is an n×n matrix, can be solved by straightforward calculations. Thus we have that

$$\Phi_f(t,0) = P^{-1}\ Y(t)\ P \quad ,$$

and we can see that $\Phi_f(t,0)$ involves nothing more complicated than exponentials of integrals of the $f_i(t)$, polynomials in the $f_i(t)$, and various combinations, products, and integrals of such quantities. This was the approach used in [1], where it was assumed that the $f_i(t)$ were mutually Gaussian random processes. Using the properties of the Gaussian distribution, it was then possible to evaluate

$$E\left\{\Phi_f(t,0)\right\} . \tag{2}$$

Since $E\{x(t)\} = E\{\Phi_f(t,0)\}\ E\{x(0)\}$, we see that it is necessary to know the quantity in (2) in order to investigate the first moment asymptotic stability of (1).

In this paper we will assume that the noise processes $f_1(t), \ldots ,f_k(t)$ are independent random processes and that each is a type of filtered Poisson process. Specifically, we assume that a noise process has the following form:

$$f(t) = \begin{cases} 0 \quad , & N_t = 0 \\ \sum_{i=1}^{N_t} U_i\ h(t,\tau_i) \quad , & N_t \geq 1 \end{cases} \tag{3}$$

where N_t is a Poisson counting process with intensity function $\lambda(t) \geq 0$,

τ_i is the i^{th} occurrence time, and the amplitudes U_i are independent and identically distributed and independent of N_t. It will be assumed that $\lambda(t)$ is integrable over bounded sets and positive on a set of positive Lebesque measure. Also, it is assumed that U_1 possesses a moment generating function. Finally, it is assumed that the impulse response, or weighting function, $h(t,\tau)$ is causal; that is, $h(t,\tau) = 0$ for $t < \tau$. Noise processes of this general type are a popular model for many physical phenomena.

The characteristic functional of the random process $f(t)$ defined in (3) is given by the following [15]:

$$C_f(g) \triangleq E\left\{\exp\left[j\int_0^t f(s)\,dg(s)\right]\right\} = \exp\left(\int_0^t \lambda(\tau)\left[\Phi\left(\int_\tau^t h(s,\tau)\,dg(s)\right)-1\right]d\tau\right), \tag{4}$$

where $\Phi(a) = E\{\exp(ja\,U_1)\}$ is the characteristic function of U_1. By the proper choice of the function $g(\cdot)$, we can use the characteristic functional in (4) to evaluate (2). We now illustrate the method with examples.

Example 1: Let $n = k = 2$, and let

$$A_0 = \begin{bmatrix} -2 & 1 \\ -1 & 0 \end{bmatrix} \qquad A_1 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \qquad A_2 = \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix} .$$

In this case we can easily check that $\mathscr{L}$ is solvable and that

$$P = \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix}$$

upper triangularizes the system. Letting $y = Px$, we have

$$\dot{y}(t) = \begin{bmatrix} -1+f_1(t) & -1+f_1(t) \\ 0 & -1+f_2(t) \end{bmatrix} y(t) .$$

Thus we find that

$$y(t) = \begin{bmatrix} a_{11}(t) & a_{12}(t) \\ 0 & a_{22}(t) \end{bmatrix} y(0) ,$$

where

$$a_{ii}(t) = \exp\left[-t+\int_0^t f_i(\tau)\,d\tau\right] , \quad i = 1, 2$$

and

$$a_{12}(t) = \int_0^t \exp\left[-t + \int_s^t f_1(\tau)\,d\tau + \int_0^s f_2(\tau)\,d\tau\right]\left(-1 + f_1(s)\right) ds .$$

Assume that both of the noise processes have the form of (3) and that for each of the noise processes the corresponding amplitudes $\{U_i\}$ have a Gaussian distribution with zero mean and unit variance. Let

$h_1(t,\tau) = \exp(\tau - t)\, u(t-\tau)$ and $h_2(t,\tau) = u(t-\tau) - u(t-\tau-1)$, where $u(\cdot)$ is the unit step function.

Using (4) we find that

$$E\left\{a_{ii}(t)\right\} = \exp\left(-t + \int_0^t \lambda_i(\tau)\left[M\left(\int_\tau^t h_i(s,\tau)\, ds\right) - 1\right] d\tau\right) ,$$

where $M(a) = \exp(a^2/2)$. Since

$$M\left(\int_\tau^t h_i(s,\tau)\, ds\right) \leq \sqrt{e} \quad \text{for all } t,\tau,$$

we have that

$$E\left\{a_{ii}(t)\right\} \leq \exp\left[-t + \left(\sqrt{e} - 1\right)\int_0^t \lambda_i(\tau)\, d\tau\right] .$$

Therefore, if

$$\int_0^\infty \lambda_i(\tau)\, d\tau < \infty \tag{5}$$

or if

$$\lambda_i(\tau) \leq \alpha \Big/ \left(\sqrt{e} - 1\right) \qquad \text{for some } \alpha < 1 , \tag{6}$$

then we see that

$$\lim_{t\to\infty} E\left\{a_{ii}(t)\right\} = 0 .$$

Now consider the off-diagonal term:

$$\begin{aligned} a_{12}(t) = &- \int_0^t \exp\left[-t + \int_s^t f_1(\tau)\, d\tau + \int_0^s f_2(\tau)\, d\tau\right] ds \\ &+ \int_0^t f_1(s) \exp\left[-t + \int_s^t f_1(\tau)\, d\tau + \int_0^s f_2(\tau)\, d\tau\right] ds . \end{aligned} \tag{7}$$

The magnitude of the expectation of the first summand in (7) can be upper bounded by

$$\int_0^t \exp\left[-t + \left(\sqrt{e} - 1\right)\int_s^t \lambda_1(\tau)\, d\tau + \left(\sqrt{e} - 1\right)\int_0^s \lambda_2(\tau)\, d\tau\right] ds . \tag{8}$$

Thus we see that if λ_1 and λ_2 each satisfy either (5) or (6) then (8) goes to zero as t approaches infinity.

To evaluate the expectation of the second summand in (7) we proceed as follows. Using (4) we find that

$$E\left\{\exp\left[r\, f_1(s) + \int_s^t f_1(\tau)\, d\tau\right]\right\} = \exp\left(\int_s^t \lambda_1(\tau)\left[M\left(r+1-e^{\tau-t}\right)-1\right]d\tau\right).$$

Thus,

$$E\left\{f_1(s)\, \exp\left[\int_s^t f_1(\tau)\, d\tau\right]\right\} = \frac{d}{dr}\, E\left\{\exp\left[r\, f_1(s) + \int_s^t f_1(\tau)\, d\tau\right]\right\}\Bigg|_{r=0}$$

$$= \exp\left[\int_s^t \lambda_1(q)\left[M\left(1-e^{q-t}\right)-1\right]dq\right] \int_s^t \lambda_1(\tau)\left(1-e^{\tau-t}\right)M\left(1-e^{\tau-t}\right)d\tau \quad .$$

Therefore,

$$\left|E\left\{f_1(s)\, \exp\left[\int_s^t f_1(\tau)\, d\tau\right]\right\}\right| \leq \sqrt{e}\int_s^t \lambda_1(\tau)d\tau\, \exp\left[\int_s^t \lambda_1(q)\left(\sqrt{e}-1\right)dq\right] . \tag{9}$$

If λ_1 satisfies (5), then (9) can be upper bounded by a constant. If λ_1 satisfies (6), then (9) can be upper bounded by $3(t-s)\exp[\alpha(t-s)]$.

Now consider

$$E\left\{\exp\left[\int_0^s f_2(\tau)\, d\tau\right]\right\}.$$

This can be upper bounded by

$$\exp\left[\left(\sqrt{e}-1\right)\int_0^s \lambda_2(\tau)\, d\tau\right] .$$

If λ_2 satisfies (5) or (6), this can be upper bounded by a constant times $\exp(\alpha s)$. Therefore, if λ_1 and λ_2 each satisfy either (5) or (6), then $E\{a_{12}(t)\}$ goes to zero as t approaches infinity, and we conclude that the system of Example 1 is asymptotically stable.

Example 2: Consider the following scalar equation:

$$\dot{x}(t) = \alpha\, x(t) + \beta\, x(t)\, f(t) \tag{10}$$

where $\beta \neq 0$. The solution is given by

$$x(t) = x(0)\, \exp\left[\alpha\, t + \beta\int_0^t f(\tau)\, d\tau\right] .$$

Assume that f(t) is a compound Poisson process; that is, in (3), $h(t,\tau) = u(t-\tau)$. Then, using (4), we have that

$$E\left\{\exp\left[\beta\int_0^t f(\tau)\, d\tau\right]\right\} = \exp\left(\int_0^t \lambda(\tau)\left(M\left[\beta\,(t-\tau)\right]-1\right)d\tau\right) ,$$

where M is the moment generating function of U_1. It is straightforward to show that if $P\{\beta U_1 > 0\} > 0$ then there exists a $\gamma > 0$ such that for $t > \tau$, $M[\beta(t-\tau)] \geq \exp[\gamma(t-\tau)]$. Assuming that $P\{\beta U_1 > 0\} > 0$, we have that

$$E\left\{\exp\left[\beta \int_0^t f(\tau)\,d\tau\right]\right\} \geq \exp\left[\int_0^t \lambda(\tau)\left(\exp\left[\gamma(t-\tau)\right] - 1\right) d\tau\right] .$$

Notice that

$$\frac{d}{dt}\left(\int_0^t \lambda(\tau)\left(\exp\left[\gamma(t-\tau)\right] - 1\right) d\tau\right) = \gamma e^{\gamma t} \int_0^t \lambda(\tau)\, e^{-\gamma\tau}\, d\tau . \tag{11}$$

Since $\lambda(\cdot)$ is positive on a set of positive Lebesque measure, we see that (11) approaches infinity as t approaches infinity. This yields the following result.

Theorem: Assume that f(t) is a compound Poisson process whose amplitudes U_i possess a moment generating function and that the intensity function $\lambda(\cdot)$ is positive on a set of positive Lebesque measure. Also, assume that $P\{\beta U_1 > 0\} > 0$. Then (10) is not asymptotically stable.

Notice that if the support of the distribution of U_1 includes both positive and negative numbers, then $P\{\beta U_1 > 0\} > 0$.

Example 3: Now consider the following n^{th} order system:

$$\dot{x}(t) = A_0\, x(t) + A_1\, x(t)\, f(t) \tag{12}$$

where A_0 and A_1 are upper triangular n×n matrices and x(t) is an n dimensional vector. Notice that the i^{th} term of the diagonal of the transition matrix $\Phi_f(t,0)$ will be of the form

$$\exp\left[\alpha_i\, t + \beta_i \int_0^t f(\tau)\,d\tau\right] , \tag{13}$$

where α_i and β_i are the i^{th} terms of the diagonals of A_0 and A_1, respectively. If the initial state x(0) is a constant vector with a one in the i^{th} position and zeros elsewhere, then the i^{th} component of x(t) will be given by (13). This results in the following corollary.

Corollary: If at least one diagonal element β of A_1 is nonzero and if f(t) satisfies the conditions of the Theorem, then (12) is not asymptotically stable.

Example 4: Consider the system (10) and let the noise process be given by (3) where

$$h(t,\tau) = \exp\left[-\gamma(t-\tau)\right] u(t-\tau) , \quad \gamma > 0 .$$

Let $M(\cdot)$ be the moment generating function of U_1. We find that

$$E\left\{\exp\left[\beta \int_0^t f(\tau)\, d\tau\right]\right\} = \exp\left[\int_0^t \lambda(\tau)\left(M\left[\frac{\beta}{\gamma}\left(1 - \exp\left[-\gamma(t-\tau)\right]\right)\right] - 1\right) d\tau\right] .$$

Assuming that $\lambda(\cdot)$ is integrable, we see that this quantity can be easily upper bounded. In this case we see that (10) is asymptotically stable if $\alpha < 0$.

Example 5: Consider the scalar equation (10). Let $f(t)$ be given by (3) and assume that h is a function of the difference of its arguments, say

$$h(t,\tau) = \tilde{h}(t-\tau) .$$

Assume that U_1 possesses the moment generating function $M(\cdot)$. Also, assume that

$$\int_0^\infty \left|\tilde{h}(t)\right| dt < \infty . \tag{14}$$

Then from (4) we see that

$$E\left\{\exp\left[\beta \int_0^t f(\tau)\, d\tau\right]\right\} = \exp\left[\int_0^t \lambda(\tau)\left(M\left[\beta \int_0^{t-\tau} \tilde{h}(s)\, ds\right] - 1\right) d\tau\right] .$$

From (14) we conclude that

$$M\left[\beta \int_0^{t-\tau} \tilde{h}(s)\, ds\right] \leq M_0 < \infty .$$

Assume that the intensity function $\lambda(\cdot)$ is such that

$$\lim_{t \to \infty} \frac{1}{t} \int_0^t \lambda(\tau)\, d\tau = 0 .$$

Then we have that

$$\left| E\left\{x(t)\right\}\right| \leq \left| E\left\{x(0)\right\}\right| \exp\left[\alpha t + M_0 \int_0^t \lambda(\tau)\, d\tau\right] .$$

Thus we see that if $\alpha < 0$, the system is asymptotically stable.

CONCLUSIONS

We have presented a method for analyzing the p^{th} moment asymptotic stability of a class of linear systems with multiplicative state noise. The class of systems considered were required to satisfy a certain Lie algebraic condition. The noise processes were taken to be a form of filtered Poisson noise. The utility of the method was illustrated with several examples. We note that these methods can be extended in a straightforward fashion to include noise processes having the form considered in this paper plus an additive independent Gaussian component.

ACKNOWLEDGEMENTS

This research was supported by the Department of Defense Joint Services Electronics Program under Contract F49620-77-C-0101 and by the National Science Foundation under Grant ENG 76-11106.

REFERENCES

1. A. S. Willsky, S. I. Marcus, and D. N. Martin, "On the stochastic stability of linear systems containing colored multiplicative noise," *IEEE Trans. Automatic Control*, vol. AC-20, Oct. 1975, pp. 711-713.

2. F. Kozin, "A survey of stability of stochastic systems," *Automatica*, vol. 5, 1969, pp. 95-112.

3. W. Wedig, "Regions of instability for a linear system with random parametric excitation," in *Proc. Symp. on Stochastic Dynamical Syst.*, (Math. Lecture Notes 294), New York: Springer-Verlag, 1972.

4. W. M. Wonham, "Random differential equations in control theory," in *Probabilistic Methods in Applied Mathematics*, vol. 2, A. T. Bharucha-Reid, ed. New York: Academic Press, 1968.

5. D. L. Kleinman, "On the stability of linear stochastic systems," *IEEE Trans. Automatic Control*, vol. AC-14, Aug. 1969, pp. 429-430.

6. R. W. Brockett, "Lie theory and control systems on spheres," *SIAM J. Appl. Math.*, vol. 25, Sept. 1973, pp. 213-225.

7. R. W. Brockett, "Lie algebras and Lie groups in control theory," in *Geometric Methods in Systems Theory*, D. Q. Mayne and R. W. Brockett, eds. The Netherlands: Reidel, 1973.

8. J. L. Willems, "Stability of higher order moments for linear stochastic systems," *Ingenieur-Archiv,* vol. 44, 1975, pp. 123-129.

9. A. S. Willsky and S. I. Marcus, "Analysis of bilinear noise models in circuits and devices," *Journal of the Franklin Institute,* vol. 301, Jan. 1976, pp. 103-122.

10. J. L. Willems, "Stability criteria for stochastic systems with colored multiplicative noise," *Acta Mechanica,* vol. 23, 1975, pp. 171-178.

11. G. L. Blankenship, "Lie theory and the moment stability problem in stochastic differential equations," *Preprints IFAC 75,* Cambridge, Mass., Aug. 1975, # 36.2.

12. M. B. Nevelson and R. Z. Khas´minskii, "Stability of a linear system with random disturbances of its parameters," *Appl. Math. Mech.,* vol. 30, 1966.

13. J. L. Willems and D. Aeyels, "An equivalence result for moment stability criteria for parametric stochastic systems and Itô equations," *Int. J. Systems Sci.,* vol. 7, 1976, pp. 577-590.

14. J. L. Willems and D. Aeyels, "Moment stability of linear stochastic systems with solvable Lie algebras," *IEEE Trans. Automatic Control,* vol. AC-21, April 1976, p. 285.

15. D. L. Snyder, *Random Point Processes,* pp. 168-169. New York: John Wiley, 1975.

GEOMETRY OF PARAMETRIC MODELS: SOME PROBABILISTIC QUESTIONS

P. S. KRISHNAPRASAD
Case Western Reserve University
Systems and Computer Engineering and Information Sciences

ABSTRACT

Input-output maps with parameters appear naturally as a consequence of certain scalings. We discuss here the role of these scalings in relation to the geometry of rational functions. These geometric questions appear to be of interest for identification problems. In this connection we pose a problem of setting up densities on the space of models and examine some solutions. Our investigations make interesting contact with analytical mechanics.

INTRODUCTION

A theory of scaling for rational functions first introduced in [1] leads naturally to the notion of i/0 maps with parameters. The first part of this paper is a collection of results on the geometry of input-output maps. We then investigate the implications of these results for identification experiments and propose a Bayesian framework for statistical inference on systems. In the second part we examine some invariants of scaling and show how these lead to densities on transfer functions.

1. PARAMETRIC MODELS

Dynamical systems with parameters have been studied extensively in connection with questions on qualitative behavior of differential equations [2]. In the context of input-output systems, transfer functions with parameters appear as a consequence of an experimenter's choice of optimum/convenient conditions. To illustrate, associated with a $g(s) \varepsilon \mathrm{Rat}(n)$, the set of proper rational functions of McMillan degree n without common factors we have the four-parameter family

$$F = \left\{ \frac{mg(\alpha s+\sigma)}{1+mk\; g(\alpha s+\sigma)} : \alpha \varepsilon R^+,\ m \varepsilon R^+,\ \sigma \varepsilon R,\ k \varepsilon R \right\}$$

Here, m and α are magnitude and frequency scalings and k is a feedback gain. The parameter σ effects exponential scaling. We make precise these ideas by defining G-scaling of a smooth manifold M by a Lie group G to be a smooth G-action, $\phi : G \times M \rightarrow M$. Points in the same orbit are then thought of as scaled versions of each other. In the present context we are interested in scalings of Rat(n) carrying a smooth manifold structure [3]. Thus, the four-parameter family F of transfer functions defined above is a typical orbit of the action of the group G_A generated by the four one-parameter scalings:

(a) $s \rightarrow \alpha s \quad ; \quad \alpha \varepsilon R^+$
(b) $s \rightarrow s + \sigma \quad ; \quad \sigma \varepsilon R$
(c) $g(s) \rightarrow mg(s) \quad ; \quad m \varepsilon R^+$
(d) $g(s) \rightarrow \dfrac{g(s)}{1 + kg(s)} \quad ; \quad k \varepsilon R$

It was indicated in [1] (see [4] for proof) that G_A is isomorphic to the group of 4 x 4 matrices of the form:

$$\begin{bmatrix} 1 & \sigma & 0 & 0 \\ 0 & \alpha & 0 & 0 \\ 0 & 0 & 1 & mk \\ 0 & 0 & 0 & m \end{bmatrix}$$

By introducing a fifth scaling, time shift,

$$\text{(e)} \quad g(s) = c(sI-A)^{-1}b \to c(sI-A)^{-1}\, e^{A\tau}b \qquad \tau \in R$$

(or equivalently)

$$ce^{At}b = w(t) \to w(t+\tau) \ ,$$

and combining this with the scalings (a), (b) and (c) above, we have another group G_B acting on Rat(n). We recall from [1] that G_B has a matrix representation.

$$\begin{bmatrix} 1 & \tau & \ell n(m) \\ 0 & \alpha & \sigma \\ 0 & 0 & 1 \end{bmatrix}$$

We are interested in the disposition of the orbits and their structure. For example, one may ask if the orbits are connected and whether all orbits have the same dimension as integral submanifolds of Rat(n). First we need a lemma.

Lemma 1: The scaling operations (a)-(e), preserve the McMillan degree. Further, all except (e) leave the number of zeros unchanged.

The proof is easy (see [4]). The requirement that there be no common factors disconnects the space Rat(n).

One of the main results in [3] was the following.

Thm 1: (Brockett) Rat(n) has (n+1) arcwise connected components. All members of a particular component have the same Cauchy index which takes values in the set $\{-n, -n+2, \ldots, n-2, n\}$.

Denoting as Rat(p,q) the connected component of Rat(p+q) of Cauchy index (p-q), we see that Lemma 1 implies that orbits of G_A and G_B lie entirely within some Rat(p,q). Further, since the groups G_A and G_B are connected Lie groups, each orbit is connected. Now, for every smooth action ϕ by a group G on a smooth manifold M we have the following: To every $m \in M$, there corresponds a C^∞ map denoted by m also of G into M defined by

$$m(g) = gm = \phi(g,m)$$

Let dm denote the corresponding derivative map from the tangent space at e (Lie algebra $\tilde{G}$) of G. Then we have the following Lie algebra homomorphism,

$$\lambda \colon \tilde{G} \to U(M) = \text{Lie algebra of } C^\infty \text{ vector fields on } M$$

$$X \to \overline{X} = \lambda X \text{ defined by } (\lambda X)\ (m) = dm(X(e))$$

The orbits of G are integral submanifolds of M of constant dimension if dm is kernel-free for all $m \in M$. This follows from the Frobenius theorem [5]. The condition on dm is the same as saying that G acts freely on M. This question was first raised in connection with G_A action on Rat(n) in [1] and there a preliminary condition was given. Complete results and proofs are in [4] and will appear elsewhere. We state,

Thm 2: The groups G_A and G_B act freely (hence effectively) on Rat(p,q) iff $|p-q| > 1$, in which case Rat(p,q) can be given the structure of a fiber bundle with structure group G_A or G_B.

The situation is complicated by the fact that the natural map $\Pi:\text{Rat}(p,q) \to \text{Rat}(p,q)/G$ for $G = G_A$ or G_B need not be a fiber map [6,7]. More precisely Rat(p,q) is a (trivial) principal G_A-bundle over Q_A

$$\mathcal{B}_A \overset{\Delta}{=} (\text{Rat}(p,q),\ \tilde{\Pi}_A,\ Q_A,\ G_A)$$

$$\text{Rat}(p,q) \tilde{=} Q_A \times G_A$$

where the base space Q_A is the set of rational functions of the form

$$g(s) = \frac{s^{m-1} + 0\cdot s^{m-2} + q_{m-3}s^{m-3} + .. + q_{\ell+1}s^{\ell+1} + 1\ s^{\ell}}{s^n + p_{n-1}s^{n-1} + \ldots. + p_1 s + p_0}$$

with $(q_{n-1}, q_{n-2}, \ldots., q_0)$ orthogonal to $(p_{n-1}, p_{n-2}, \ldots, p_0)$. A similar smooth orbit-canonical form for G_B exists (see [4] for details).

At the level of realizations also, we can exhibit such structure. Denoting as $\hat{R}$ the set of all minimal triples [A, b, c]. We have a trivial bundle with structure group (fiber) $G\ell(n)$,

$$(\hat{R},\ \Pi,\ \text{Rat}(n),\ G\ell(n))$$

$$\hat{R} \cong \text{Rat}(n) \times G\ell(n)$$

with

$$\Pi([A,\ b,\ c]) = c(sI-A)^{-1}b.$$

We see that parametric sets of i/o maps can be given a precise geometric structure under some conditions. Since the scalings of this section are dependent on an experimenter's free choice, there appears to be an element of ambiguity with regard to the outcome of identification experiments and tests of hypotheses about systems. We examine this and related questions below.

2. IMPLICATIONS FOR SYSTEM IDENTIFICATION

To focus ideas, consider the concrete problem of identifying the system,

$$(*)\qquad \begin{aligned} x_{t+1} &= Ax_t + bu_t \\ y_t &= cx_t \end{aligned}$$

where $x_t \in \mathcal{R}^n$, $A \in \mathcal{R}^{n\times n}$, $b \in \mathcal{R}^{n\times 1}$, $c \in \mathcal{R}^{1\times n}$ and $\{u_t\}$ is discrete-time standard Gaussian white noise. To identify the transfer function $g(z) = c(zI-A)^{-1}b$, let $\mathcal{I}(\theta, \hat{\theta}, y, T)$ be the support function or identification criterion, i.e. if θ represents the true transfer function and $\hat{\theta}$ an estimate then $I(\theta, \hat{\theta}, y, T)$ is the measure of support for $\hat{\theta}$, given by a realization $y(\cdot)$ of length T. A commonly used support function is the log-likelihood function. To eliminate ambiguity due to scalings, one asks for invariance in the sense of [1], i.e.

$$I(\theta, \hat{\theta}, y, T) = I(g(\theta), g(\hat{\theta}), g(y), g(T)) \text{ wp1} \quad \forall g \in G$$

where G is a (scaling) Lie group acting on Rat(n).

This question was examined in detail in [1] in relation to (*), using the asymptotic expression of support,

$$V(\theta,\hat{\theta}) = \frac{1}{2\pi i} \oint_C \frac{g(z)g(z^{-1})}{\hat{g}(z)\,\hat{g}(z^{-1})} \frac{dz}{z}$$

where C is the circle, $|z| = 1$. Let us denote the transfer function as

$$g(z) = \frac{1 + q_{m-1} z^{-1} + \ldots . + q_0 z^{-m}}{1 + p_{n-1} z^{-1} + \ldots . + p_0 z^{-n}}$$

The question of main interest to us is the effect of the geometry of Rat(n) on the identification criterion and V is a convenient object for this study.

Firstly, V is insensitive to common factors. Secondly, in a variety of practical problems, it is hard to arrive at a proper guess of the bidegree (n,m) where n = number of poles and m = number of zeros. As a result, some trial and error or more formally hypothesis-testing is involved. Since an estimate $\hat{g}(z)$ that achieves the maximum of log-likelihood or equivalently minimum of V is often desirable [8], it is necessary to understand the geometry of the set of minima of $V = V(\hat{q},\hat{p})$ as a function of

$$\hat{g}(z) = \frac{\hat{q}(z)}{\hat{p}(z)} = \frac{1 + \hat{q}_{\hat{m}-1} z^{-1} + \ldots . + \hat{q}_0 z^{-\hat{m}}}{1 + \hat{p}_{\hat{n}-1} z^{-1} + \ldots . + \hat{p}_0 z^{-\hat{n}}}$$

We shall summarize below some results due to Aström and Söderström [9]. First some definitions.

Def^{n}1: Let $DRat_{SM}(n;m) \triangleq$ set of all rational functions without common factors with McMillan degree = n and number of zeros = $m \leq n$ and all poles and zeros lying in the open unit disk.

Def^{n}2: Let $\overline{DRat}_{SM}(n;m) \supset DRat_{SM}(n;m)$ be the set obtained by removing the restriction on common factors in Definition 1. The following is a version of the reults in [9].

Thm 3: (Aström and Söderström): Let

$$g(z) = \frac{1 + q_{m-1} z^{-1} + \ldots + q_0 z^{-m}}{1 + p_{n-1} z^{-1} + \ldots + p_0 z^{-n}}$$

$q_0 \neq 0$, $p_0 \neq 0$, $n \geq m$, belong to $DRat_{SM}(n;m)$. Consider the map, $V:\overline{DRat}_{SM}(\hat{n};\hat{m}) \to R^1$ defined by

$$\hat{g}(z) \to \frac{1}{2\pi i} \oint_C \frac{g(z)\, g(z^{-1})}{\hat{g}(z)\, \hat{g}(z^{-1})} \frac{dz}{z} = V$$

Then

$$V(\hat{q},\hat{p}) \geqq 1 \quad \text{and}$$

$$V(\hat{q},\hat{p}) = 1 \quad \text{iff}$$

$$(*) \qquad q(z)\, \hat{p}(z) = \hat{q}(z)\, p(z).$$

Further,

(a) Equ (*) has a solution iff $\min(\hat{n}-n,\ \hat{m}-m) \geqq 0$.

(b) If $\hat{n}=n$ and $\hat{m}\geqq m$ or $\hat{n}\geqq n$ and $\hat{m}=m$, then (*) has a unique solution which is the global minimum of V on $\overline{\mathrm{D}}\mathrm{Rat}_{SM}(\hat{n};\hat{m})$.

(c) If $\min(\hat{n}-n,\ \hat{m}-m) \geqq 0$, then the only critical pts. of V are the solutions of (*).

What we notice here is that in searching for a maximum likelihood estimate in $\overline{\mathrm{D}}\mathrm{Rat}_{SM}(\hat{n};\hat{m})$with $\hat{n}>n$ and m>m we never find the best approximation within $\mathrm{DRat}_{SM}(\hat{n};\hat{m})$. For this it is necessary to fix $(\hat{n};\hat{m})$ and let $\mathrm{DRat}_{SM}(\hat{n};\hat{m})$ be the manifold on which to search for min(V). This is also required since for $\hat{n}<n$ or $\hat{m}<m$, critical points of V need not be minima. Further, the Fisher-Riemann metric [1] defined by

$$F_{pq} = \left. \frac{\partial^2 V}{\partial \hat{\theta}^p\, \partial \hat{\theta}^q} \right|_{\hat{\theta} = \theta},$$

is degenerate at common factors (this is a consequence of the Morse lemma [10] which requires nondegenerate critical pts. to be isolated) causing unidentifiability. Our solution to the problem is to treat $\mathrm{DRat}_{SM}(n;m)$ or Rat(n) as the parameter space and carry out Bayesian inference as follows:

Maximize (Posterior Support)

Where Posterior support = log (likelihood) + log (p(g)) where p(g) the prior density satisfies the nondegneracy condition that it is nonzero on Rat(n) and vanishes at commonfactors (i.e. $p(g) \to 0$ as resultant $(g) \to 0$). Any algorithm to search for the maximum of the posterior support will remain within connected components because of the nondegeneracy condition. One natural candidate for a prior is the volume element associated with the Fisher-Riemann metric. Variational priors in general [4, 11] belong to this category. Densities that admit invariant Bayesian inference are the subject of the rest of this paper.

3. INVARIANTS OF SCALINGS

Let

$$g(s) = \frac{q(s)}{p(s)} = \frac{q_{n-1}\, s^{n-1} + q_{n-2}\, s^{n-2} + .. + q_1 s + q_0}{s^n + p_{n-1}\, s^{n-1} + ... + p_1 s + p_0}$$

Consider the Bezoutian form $B(q,p) = q(x)p(y) - q(y)p(x)$. Since for

$x = y$, $B(q,p) = 0$, $(x-y)$ divides the above form. Therefore, we have

$$\tilde{R}(q,p;\ x,y) = (q(x)\,p(y) - q(y)\,p(x))/(x-y)$$
$$= \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ik}\, x^{i-1}\, y^{k-1} \qquad c_{ik} = c_{ki}$$

Then it can be shown that the resultant of q and p is given by [12].

$$\underset{\sim}{R} = \mathrm{Res}(q,p) = \det[c_{ik}]$$

Thus $g(s) \in \mathrm{Rat}(n)$ iff the quadratic form $[c_{ik}]$ is nondegenerate. Firstly we view the resultant as a nonvanishing function on Rat(n) (a differential form of degree 0).

Def^{n}3: A differential form ω on a manifold M is said to be a relative invariant under the action of a group G on M if there exists a function $\psi : G \to R^1$ such that $\omega(gm) = \psi(g)\cdot\omega(m)$ $\forall g \in G$ and $m \in M$. Further, it is an absolute invariant if $\psi(g) = 1$ $g \in G$. We have,

Lemma 2: The resultant R is a relative invariant of the scaling group G_A satisfying the following transformations:

$$
\begin{array}{llll}
(1) & s \to \alpha s & (\alpha \in R^+)\ ; & \underset{\sim}{R} \to \alpha^{-n^2} \underset{\sim}{R} \\
(2) & s \to s + \sigma & (\sigma \in R)\ ; & \underset{\sim}{R} \to \underset{\sim}{R} \\
(3) & g(s) \to m\, g(s) & (m \in R^+)\ ; & \underset{\sim}{R} \to m^n \underset{\sim}{R} \\
(4) & g(s) \to \dfrac{g(s)}{1 + k\, g(s)} & (k \in R)\ ; & \underset{\sim}{R} \to \underset{\sim}{R}
\end{array}
$$

Remark: The resultant is not even a relative invariant of the shift $w(t) \to w(t + \tau)$. To see this consider the example:

$$g(s) = \frac{\alpha}{s + \lambda}\ ;\ \underset{\sim}{R} = \alpha. \quad \text{Under } t \to t + \tau,\ \underset{\sim}{R} \to e^{-\lambda\tau}\underset{\sim}{R}.$$

To understand what the other invariants are, consider the infinitesimal feedback scaling (generates $g(s) \to \dfrac{g(s)}{1-tg(s)}$).

$$(+) \qquad \frac{dq_i}{dt} = 0;\ \frac{dp_i}{dt} = -q_i \qquad i=0,\ 1,\ 2,\ ..,\ n-1\ .$$

This is a Hamiltonian system in R^{2n} with

$$H = H(q,p) = \sum_{i=0}^{n-1} q_i^{\ 2}/2$$

To describe this in Rat(p,q), recall that R^{2n} carries a canonical symplectic structure [5],

$$\omega = \Sigma\, dq_i \wedge dp_i$$

and if f denotes the natural imbedding $f : \mathrm{Rat}(n) \to R^{2n}$ then $\omega_F = f^*\omega$ is the natural pullback closed 2-form on Rat(p,q). Denoting by,

$$\lambda_F = \Sigma - q_i \frac{\partial}{\partial p_i}$$

the feedback vectorfield on Rat(p,q) we see that (+) is a statement of the invariance,

$$\text{(Lie derivative wrt}\lambda_F) \quad D_{\lambda_F}\,\omega_F = 0$$

This immediately implies,

$$(++) \qquad D_{\lambda_F}\,\Omega = 0$$

where $\Omega = (\omega_F)^{\Lambda n}$ is the canonical 2n form. Thus, we see from (++) that Rat(n) carries a canonical smooth invariant (under feedback) measure with density Ω. This definitely does not satisfy the nondegeneracy condition of Section 3. However the associated 2n form

$$\Omega_1 = \underset{\sim}{R} \cdot \Omega$$

is a λ_F-invariant volume element (from Lemma 2) and also satisfies the nondegeneracy condition.

The symplectic structure ω_F is not the only possible one. Consider, for example,

$$g(s) = \sum_{i=1}^{n} \frac{e^{\alpha_i}}{s + \lambda_i} \;\varepsilon\; \text{Rat}(n,o)$$

$\lambda_i \neq \lambda_j$ for $i \neq j$ and α_i real. Under the shift $t \to t + \tau$

$$g(s) \to \Sigma \frac{e^{\alpha_i - \lambda_i \tau}}{s + \lambda_i} \; .$$

The corresponding infinitesimal representation

$$(+) \qquad \frac{d\alpha_i}{dt} = \lambda_i \; ; \; \frac{d\lambda_i}{dt} = 0 \quad t = -\tau$$

This is also a Hamiltonian system with $H = \frac{1}{2}\sum_i \lambda_i^2$.

The associated (real) symplectic structure ω_s on Rat(n,o) is

$\omega_s = \Sigma\, d\alpha_i \wedge d\lambda_i$ and the canonical smooth shift-invariant density is given by $\Omega_s = (\omega_s)^{\Lambda n}$

Remark: (1) ω_s and ω_F are distinct in that there is no symplectic diffeomorphism carrying one to the other.

(2) The flow (+) and the form ω_s coincide with Moser's representation of the Toda lattice [13] and this suggests natural generalizations of isospectral deformations.

In this section we have seen two densities Ω, Ω_1 on Rat(n) associated with a natural symplectic structure. We also noted that the Fisher-Riemann metric given rise to a smooth density. In the next section these ideas are unified in one common setting.

4. INTEGRAL GEOMETRY ON Rat(n)

The general approach of integral geometry is as follows: Let M be a manifold of dimension n.

(a) Find $\omega_1, \omega_2, \ldots, \omega_n$ a set of independent differential 1-forms on M, which are invariant ωrt a Lie group G acting on M, i.e. if X is an element belonging to the Lie algebra $\tilde{G}$ of G mapped into an element $\overline{X}$ ε U(M) the set of C^∞ vectorfields on M then

$$D_{\overline{X}}\, \omega_i = 0 \quad i = 1, 2, .. n .$$

(b) Then, $\Omega = \bigwedge_{i=1}^{n} \omega_i$ is an invariant density.

Unless M is compact such densities are not probability densities. However, the set function $P(Q/R) = \int_{Q\cap R} \omega \Big/ \int_R \omega$ defines a conditional probability structure, where (Q,α) and (R,β) belong to some G-atlas [5] on M. We then have the following natural generalization of a conditional Poisson process:

$$(1) \quad P_n(Q/R) = \exp\Big(-\int_{Q\cap R}\omega\Big) \cdot \Big(\int_{Q\cap R}\omega\Big)^n / n! \quad n=0, 1, 2, ..$$

Where $P_n(Q/R)$ denotes the probability of n Poisson events occuring in the patch (Q,α) with the conditioning that the probability of an event occuring outside the patch (R,β) is zero. If a Riemannian structure is specified on M, then a 1-parameter family of probability densities on M is defined by the fundamental solution to the diffusion equation:

$$(2) \qquad \frac{\partial u}{\partial t} = \Delta u$$

Where the Laplace-Beltrami operator,

$$(3) \qquad \Delta = \frac{1}{\sqrt{g}} \frac{\partial}{\partial x^p} \left\{ g^{pq} \sqrt{g} \frac{\partial}{\partial x^q} \right\}$$

Where $g = |\det(g_{pq})|$; $g^{pq} g_{qr} = \delta^p_r$. In general it is hard to find (g_{pq}) admitting a specified group (such as G_A or G_B) as a group of isometries. We consider several cases below.

(a) Probability Density on Rat(n,o): Recall that any g(s) ε Rat (n,o) has a (symmetric) realization [A,b,b'] where A=A' has distinct (real) eigenualues and [A,b] is a cyclic pair. If [F,g,g'] is another such then there is an unique M ε O(n) such that F = MAM'; g = Mb. Denoting as $\hat{R}_s$ the set of such realizations we have on $\hat{R}_s$ the metric.

$$(4) \quad ds^2 = \mathrm{tr}(dA^2) + \langle db, db \rangle$$

invariant under $O(n) \times \{\pm 1\} \times R^{n(n+3)/2}$ represented by A → MAM'; b → Mb; A → A+R; b → b+ℓ; $R=R^T$, ℓ ε R^n and M ε O(n). Upto scale factor, the corresponding

$$(5) \qquad \Delta = \frac{1}{2} \sum_{i \leq n} \frac{\partial^2}{\partial a_{ii}^2} + \frac{1}{4} \sum_{i<j} \frac{\partial^2}{\partial a_{ij}^2} + \frac{1}{2} \sum_i \frac{\partial^2}{\partial b_i^2}$$

The corresponding density is given by

$$(6) \qquad p(A,b) \sim \exp\left(\frac{-\mathrm{tr}A^2}{2t}\right) \quad \exp\left(-\frac{<b,b>}{2t}\right)$$

which induces the following density on poles

$$(7) \qquad p(\lambda_1, \lambda_2, \ldots, \lambda_n) \sim \exp\left(-\frac{1}{2}\sum_i \lambda_i^2\right) . \quad \prod_{i<j} |\lambda_i - \lambda_j| \qquad \lambda_i \in R$$

Note that it vanishes at repeated poles as it should since in Rat(n,o), repeated poles are inadmissible. However, the cyclicity of [A,b] is not taken into account. The density (7) first arose in Physics in the study of statistics of energy levels and is discussed by McKean [14]). A more natural metric from a system theoretic viewpoint appears in treating the general case.

(b) Riemannian Structure on Minimal Realizations: Let $\hat{R}$ denote the set of all minimal triples [A,b,c]. Let the transfer function $g(s) = c(sI-A)^{-1}b$. Further, let M and N be respectively the controllability and observability matrices. $M = [b, Ab, .., A^{n-1}b]$; $N = [c; cA; \ldots; cA^{n-1}]$. Under R^n change of basis $x \to Px$, $P \in G\ell(n;R)$, we have, $A \to PAP^{-1}$; $b \to Pb$; $c \to cP^{-1}$; $M \to PM$ and $N \to NP^{-1}$. Consider the quadratic differential form,

$$(8) \qquad ds^2 = \mathrm{tr}(dAdA') + <M'dc, M'dc> + <Ndb, Ndb>$$

Equ. (8) defines a $G\ell(n)$ invariant Riemannian metric on $\hat{R}$ and the associated volume element is

$$(9) \qquad d\mu = \sqrt{g} \; \prod_{ij} dA_{ij} \prod_k db_k \prod_\ell dc_\ell$$

where g = det(MM') det(NN') = det (H) where H is the system Hankel matrix. But det(H) = |Res(q,p)| (see [4]). Hence, the density associated with (8) satisfies the nondegeneracy condition and this is true for the solutions of the diffusion equation also. In [4] the projection of the corresponding Laplace operator onto Rat(n) is considered.

(c) Riemannian Structure on Rat(n): Let N be as defined in part (b) above. Consider the set of 2n differential forms of degree 1, defined by

$$\omega_i = dq_{i-1} \qquad i=1, 2, \ldots, n$$

$\omega_{n+1} \cdots \omega_{2n}$ defined by

$$\begin{bmatrix} \omega_1 \\ \cdot \\ \cdot \\ \cdot \\ \omega_n \end{bmatrix} = N \cdot \begin{bmatrix} dp_0 \\ dp_1 \\ \cdot \\ dp_{n-1} \end{bmatrix}$$

As a consequence of observability the $\{\omega_i\}$ span $\Omega^1(\mathrm{Rat}(n))$ and a metric $ds^2 = \omega_i \cdot \omega_i$ is available. The corresponding volume element $\omega = \underset{\sim}{R} \bigwedge_i dq_i \bigwedge_i dp_i$. The interesting thing is that on the set of all pole models of the form $g(s) = 1 / (s^n + p_{n-1} s^{n-1} + .. + p_0)$, ds^2 above is flat. Further ω is a relative invariant under G_A.

5. CONCLUSIONS

We have described here some preliminary results on geometric methods in identification.

ACKNOWLEDGEMENT

This work was supported in part by the U.S. Office of Naval Research under the Joint Services Electronic Program, Contract N00014-75-C-0648 and the Army Research Office under Grant DAAG29-76-C-0139. This work would not have been possible without Professor R. W. Brockett's kind help and encouragement.

REFERENCES

1. R. W. Brockett, P. S. Krishnaprasad; "Scaling Rational Functions and Linear System Identification"; Proceedings Conference Information Sciences and Systems, pp 501-506; 1977.
2. S. Lefschetz; "Differential Equations: Geometric Theory"; 2nd edition, InterScience, New York; 1963.
3. R. W. Brockett; "Some Geometric Questions in the Theory of Linear Systems"; I.E.E.E. Transactions on Automatic Control, vol. AC-21, pp 449-455; August, 1976.
4. P. S. Krishnaprasad; "Geometry of Minimal Systems and the Identification Problem"; Ph.D. Thesis; Harvard University; 1977.
5. R. L. Bishop, R. J. Crittendon; "Geometry of Manifolds"; Academic Press, New York; 1964.
6. H. J. Sussman; "On Quotients of Manifolds: A Generalization of the Closed Subgroup Theorem"; Bulletin A.M.S., vol. 80, no. 3, pp. 573-575; May, 1974.
7. C. J. Earle, J. Eells, Jr.; "Foliations and Fibrations"; J. Differential Geometry, 1, pp. 33-41; 1967.
8. R. A. Fisher; "Contributions to Mathematical Statistics"; Wiley, New York; 1950.
9. K-J Astrom, T. Soderstrom; "Uniqueness of the Maximum Likelihood Estimates of Parameters from ARMA Model"; I.E.E.E. Transactions on Automatic Control, vol. 19, pp. 769-773; 1974.
10. J. W. Milnor; Morse Theory: Annals of Math Studies"; Princeton University Press, Princeton; 1969.
11. R. L. Kashyap; "Prior Probability and Uncertainty"; I.E.E.E. Transactions on Inf. Th., vol. IT-17, no. 6, pp. 641-650; November, 1971.
12. M. Bocher; "Introduction to Higher Algebra"; Dover, New York; 1964.
13. J. Moser; "Dynamical Systems Theory and Applications - Lecture Notes in Physics"; Springer-Verlag, Berlin, no. 38, pp. 467-498; 1975.
14. H. P. McKean; "Stochastic Integrals"; Academic, New York; 1969.

A REPRESENTATION THEOREM FOR LINEAR DIFFERENTIAL EQUATIONS WITH MARKOVIAN COEFFICIENTS

ROGER W. BROCKETT*
Division of Applied Sciences
Harvard University
Cambridge, MA. 02138

GILMER L. BLANKENSHIP**
Division of Electric Energy Systems
Energy Researcn and Development Administration
Washington, D.C. 20545

ABSTRACT

First we show that any finite-state continuous-time Markov processes can be modeled by a stochastic differential equation of the form

$$dz = \sum_{i=1}^{\delta} \phi_i(z)dN_i$$

where the N_i are standard Poisson counting processes and z(t) satisfies a polynomial identity $p[z(t)] = 0$. This representation is then used together with the general Ito differentiation rule to show that Ito equations of the form

$$dx = A(z)x(t)dt + \sum_{i=1}^{\gamma} B_i(z)x(t)dw_i + c(z)dt + \sum_{i=\gamma+1}^{\beta} d_i(z)dw_i$$

with z a finite-state, continuous-time Markov process can be studied by studying equations in a higher dimensional space of the form

$$d\tilde{x} = \tilde{A}xdt + \sum_{i=1}^{\beta} \tilde{B}_i\tilde{x}dw_i + \sum_{i=1}^{\delta} \tilde{C}_i\tilde{x}dN_i$$

where the $\tilde{A}$, $\tilde{B}$, and $\tilde{C}$ are now independent of z. The statistical properties of the solutions of this latter equation -- as well as those of any equation derived from it via tensoring -- are easily reduced to the study of the properties of linear ordinary differential equations and in this way we get an extremely effective tool for analyzing linear stochastic differential equations with an interesting class of random coefficients. Models of this form have been widely used in fault detection, adaptive filtering and stochastic control and consequently these methods should find considerable application.

1. INTRODUCTION

Continuous-time, finite-state Markov processes (CTFS Markov process) are useful models for a wide variety of stochastic phenomena. Random telegraph waves, radioactive decay and reliability analysis suggest three different areas of application which come immediately to mind. In stochastic control, filtering, etc. one is often interested in differential equations of the general form

---

* Partially supported by the U.S. Office of Naval Research under the Joint Services Electronics Program Contract N00014-75-C-0648 and the Army Research Office under Grant DAAG29-76-C-0139.

** On leave from Systems Engineering Department, Case Western Reserve University, Cleveland, Ohio 44106. Partially supported by National Science Foundation Grant ENG75-08613.

$$dx = f(x,z,t)dt + \sum_{i=1}^{\beta} g_i(x,z,t)dw_i \tag{N}$$

where z is a CTFS Markov process and the w_i are independent Wiener processes.

It is standard to describe a CTFS Markov process by the infinitesimal transition probabilities $\{a_{ij}\}$ defined so that

$$\dot{p} = Ap \tag{K}$$

where p_i is the probability that $z=z_i$, the ith state. In this paper we develop and use a different representation for CTFS Markov processes based on stochastic differential equations driven by Poisson counters. The main advantages of this representation when dealing with dynamical problems are:

(a) The evolution of the dynamical system and that of the CTFS Markov process are both described using the same formalism.

(b) By using the appropriate differentiation rules one can, in a completely routine way, arrive at differential equations for any polynomial function of the state.

(c) By using standard results from stochastic differential equation theory we can derive differential equations for the moments and generalized covariances and determine easily to what extent these are solvable.

Section 2 contains the basic differential equation representation for CTFS Markov processes. The representation is given in terms of a number of independent Poisson counting processes $N_i(t)$ whose rates determine the transition probabilities. In section 3 we consider the application of this result to stochastic differential equations and give a more general representation theorem which makes it possible to use the Ito calculus together with basic properties of the expectation operator to study the statistical properties of the solutions of linear stochastic differential equations whose right hand sides depend on FSCT Markov processes. In section 4 we describe how equations of this form behave under tensoring and show how to derive a linear differential equation for any polynomial functions of the state. In the last section we give a number of examples.

We would like to thank Adrian Segall of the Technion for some helpful suggestions at an early state of this work.

2. FSCT MARKOV PROCESSES

By a standard Poisson counter (SPC) of rate λ we understand a denumerable-state jump process which takes on integer values, is right continuous, non-decreasing, and has the property that for t > s

$$p(N(t) = r+k | N(s) = r) = (\lambda^k/k!)e^{-\lambda(t-s)}$$

We denote such processes by N(t). Stochastic integrals with respect to dN(t) are well studied and in this section we will be interested in using this theory in connection with differential equations of the form

$$dz(t) = \sum_{i=1}^{\delta} \phi_i[z(t)]dN_i(t) \tag{M}$$

where $N_i(t)$ are independent SPC's. We interpret all integrals and differential equations in the sense of K. Ito. His study [1] included as special cases integration with respect to differentials of Wiener processes, SPC's and linear combinations of these.

Now suppose that s is an FSCT Markov process with s(t) taking on values is $S = \{s_1, s_2, \dots s_m\}$. Let $p_i(t)$ be the probability that $s = s_i$ at time t and let $\{a_{ij}\}$ be the infinitesimal transition probabilities so that the Kolmogorov equation is equation (K) of the introduction. We now show how to associate with every one to one mapping ψ of S into the complex numbers, a stochastic differential equation (M) having the properties

(a) if $z(0) \varepsilon Z = \psi(S)$ then $z(t) \varepsilon Z$ for all $t > 0$.

(b) if $z(0) \varepsilon Z$ the transition probabilities for the z-process are identical to those of the s-process, i.e. if ρ_i is the probability that $z(t) = z_i = \psi(s_i)$ then $\dot{\rho} = \rho A$.

The construction goes as follows. We let ϕ_i be polynomials with the following properties:

$$\phi_i = \begin{cases} 0 & ; \quad z \neq z_1 \\ z_i - z_1; & z = z_1 \end{cases} \qquad 1 \leqslant i \leqslant m-1$$

$$\phi_i = \begin{cases} 0 & ; \quad z \neq z_2 \\ z_{i-m-1} - z_2; & z = z_2 \end{cases} \qquad m \leqslant i \leqslant 2m-2$$

$$\cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot$$

$$\phi_i = \begin{cases} 0 & ; \quad z \neq z_m \\ z_{i-m(m-2)} - z_m; & z = z_m \end{cases} \qquad (m-1)^2+1 \leqslant i \leqslant m(m-1)$$

For $1 \leqslant i \leqslant m(m-1)$ we let N_i be an SPC process with rate λ_i given by

$$\lambda_i = a_{1+i,1} \quad ; \qquad 1 \leqslant i \leqslant m-1$$
$$\lambda_i = a_{1+i-m-1,2} \quad ; \qquad m \leqslant i \leqslant 2m-2$$
$$\cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot$$
$$\lambda_i = a_{1+i-(m-1)^2,m}; \qquad (m-1)^2+1 \leqslant i \leqslant m(m-1)$$

Now consider the evolution of the z-process. If $z(0) = z_j$ then all but m-1 of the ϕ_i's are zero and hence the behavior of all but m-1 of the SPC's is irrelevant until z(t) jumps. Those ϕ_i's which are not zero take on values in the set $\{(z_k - z_j)\}$ for $k \neq j$ hence the next state will be z_k if the coefficient of $(z_k - z_i)$ counts before any other counter in that block does. However the rates of the counters are selected in such a way as to make the probability of this counter incrementing in time ΔT equal to $a_{ki}\Delta T + o(\Delta T)$ as required. (Note that the probability of two counters incrementing simultaneously is zero.) The ϕ_i can be taken to be polynomials of degree m-1 because they need only interpolate correctly at m points. Subject to this restriction the ϕ_i are unique. We summarize this argument with a theorem.

Theorem 1: Given a finite state, continuous time jump process which takes on values in $S = \{s_1, s_2, \dots, s_m\}$; there exists for any bijection ψ: $S \to Z \subset \mathbb{C}$ a unique set of m(m-1) degree m-1 polynomials ϕ_i and a set of m(m-1) standard Poisson counters with appropriate rates such that for $z(0) \varepsilon Z$ the solution of the stochastic differential equation (M) evolves in Z and has the same transition probabilities as s does.

We illustrate with an example. Let the infinitesimal transition probabilities be given by

$$\begin{bmatrix} \dot{p}_1 \\ \dot{p}_2 \\ \dot{p}_3 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 \\ 1 & -2 & 2 \\ 0 & 2 & -3 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}$$

We pick (arbitrarily) for Z the points -1, 0, 1 and note that whereas the general procedure calls for 6 SPC's we need only 4 here because $\alpha_{31}=\alpha_{12}=0$. Let N_1,N_2,N_3, and N_4 have rates 1, 2, 1 and 2 respectively. Let ϕ_1,ϕ_2, ϕ_3 and ϕ_4 be given by

$$\phi_1 = \frac{1}{2}z(z-1)$$

$$\phi_2 = -(z-1)(z+1)$$

$$\phi_3 = -\frac{1}{2}z(z+1)$$

$$\phi_4 = -z(z+1)$$

It is easily verified that with these choices the solution of

$$dz = \sum_{i=1}^{4} \phi_i(z)dN_i$$

evolves in the set $\{-1,0,1\}$, if z(0) is in this set, and that it has the appropriate transition probabilities.

Remark 1: It might seem surprising that we have used m(m-1) independent counting processes in the general construction of a model for an m-state FSCT Markov process. This is, however, quite reasonable when we consider that the set of all m by m infinitesimally stochastic matrices is an m(m-1) dimensional manifold (with boundary) and the rates of the counters are the only mechanism we have mirror this number of degrees of freedom. Thus, in general, fewer counters will not suffice.

Remark 2: If $Z = \{z_1,z_2,\ldots,z_m\}$ then it is convenient to introduce $p(z) = (z-z_1)(z-z_2)\ldots(z-z_m)$ and think of the z equation as jumping among the zeros of p(z). This point of view emphasizes that p[z(t)] vanishes identically if p[z(0)] is zero. We have no interest here in initial data for the z equation such that $p[z(0)] \neq 0$.

We complete this section with the observation that equation (M) can be imbedded in a set of linear equations. Since this will be useful later we describe it in detail. Consider the vector

$$\tilde{z} = \begin{bmatrix} z^1 \\ z^2 \\ \vdots \\ z^{m-1} \end{bmatrix}$$

We observe that $d(z^k)$ can be expressed as

$$dz^k = \sum_{i=1}^{n} \phi_i^k(z)dN_i$$

where the $\phi_i^k(z_\ell) \overset{\text{def}}{=} (\phi_i(z_\ell))^k-(z_\ell)^k$. We ask that $\phi_i^k(z) = \phi_i^k(z)$ modulo $p(z)$ be expressed as a polynomial of degree $m-1$. Thus we can also find matrices M_i, independent of $\tilde{z}$, such that

$$d\tilde{z} = \sum_{i=1}^{\delta} M_i\tilde{z}dN_i \tag{MM}$$

We can think of z as jumping among the columns of the Vandermonde matrix

$$V = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_m \\ \vdots & \vdots & & \vdots \\ z_1^{m-1} & z_2^{m-1} & \cdots & z_m^{m-1} \end{bmatrix}$$

Equation (MM) allows us to recover the Kolmogorov equation. By re-writing (MM) as

$$d\tilde{z} = \sum_{i=1}^{\delta} M_i\tilde{z}(dN_i-\lambda_i dt) + M\tilde{z}dt; \quad M = \sum_{i=1}^{\delta} M_i\lambda_i$$

and then taking expectations, we obtain

$$\frac{d}{dt}\mathscr{E}\tilde{z} = M\mathscr{E}\tilde{z}$$

which is equivalent to equation (K). Stated somewhat differently, if we introduce $V^{-1}z = x$ in place of z then the variable x jumps among the extreme points of the standard simplex in $\mathbb{R}^m$ and

$$\frac{d}{dt}\mathscr{E}x = V^{-1}MV\mathscr{E}x$$

is exactly equation (K). This description is used in [3].

3. A CLASS OF STOCHASTIC DIFFERENTIAL EQUATIONS

Consider the stochastic differential equation (N) of the introduction. For each possible value of z we assume that $f(x,z_k)$ and $g_i(x,z_k)$ are such that the resulting Ito equation admits a unique solution. Since on any finite interval there is, with probability one, only a finite number of changes in z, we can piece together a solution to this equation and it will be unique. Using theorem 1 we can model the FSCT Markov process by equation (M) and find polynomials $\chi_i(z)$ such that equation (M) together with

$$dx(t) = \sum_{i=1}^{m} \chi_i(z)f_i(x) + \sum_{i=1}^{m}\sum_{j=1}^{\beta} \chi_i(z)g_{ij}(x)dw_j$$

models the (x,z) process.

The remainder of the paper is devoted to a particular class of problems for which this approach is especially effective. The differential equations are linear in x but have FSCT Markov coefficients, i.e. we consider equation (M) paired with

$$dx = A(z)xdt + \sum_{i=1}^{\gamma} B_i(z)xdw_i + c(z)dt + \sum_{i=\gamma+1}^{\beta} d_i(z)dw_i \tag{L}$$

The following theorem describes an extremely convenient imbedding of this equation into an equation in which the coefficient processes are independent increment processes. We always assume that the counting processes and the Wiener processes are independent.

Theorem 2: If (x,z) satisfies equations (L) and (M) then there exists $\tilde{x}$, z-independent matrices $\tilde{A}$, $\tilde{B}_i$, $\tilde{C}_i$ and a linear map R such that

$$d\tilde{x} = \tilde{A}\tilde{x}dt + \sum_{i=1}^{\beta} \tilde{B}_i\tilde{x}dw_i + \sum_{i=1}^{\delta} \tilde{C}_i\tilde{x}dN_i; \quad x = R\tilde{x} \tag{T}$$

Moreover if λ_i is the rate of the SPC N_i then

$$\frac{d}{dt}\mathcal{E}\tilde{x} = (\tilde{A} + \sum_{i=1}^{\delta} C_i\lambda_i)\mathcal{E}\tilde{x}; \quad x = R\mathcal{E}\tilde{x}$$

Proof: We introduce the vector $\begin{bmatrix}1\\x\end{bmatrix}$ so that we can write equation (L) in linear homogeneous form

$$d\begin{bmatrix}1\\x\end{bmatrix} = A^*\begin{bmatrix}1\\x\end{bmatrix}dt + \sum_{i=1}^{\beta} B_i^*\begin{bmatrix}1\\x\end{bmatrix}dw_i + \sum_{i=1}^{\delta} C_i^*\begin{bmatrix}1\\x\end{bmatrix}dN_i \tag{L'}$$

Now introduce the tensor product (recall $\tilde{z}$ from section (2))

$$\begin{bmatrix}1\\x\end{bmatrix} \otimes \tilde{z} = \tilde{x} = \begin{bmatrix} \begin{bmatrix}1\\x\end{bmatrix} \\ z\begin{bmatrix}1\\x\end{bmatrix} \\ z^{m-1}\begin{bmatrix}1\\x\end{bmatrix} \end{bmatrix}$$

By routine use of the general Ito differentiation rule together with equations (L) and (MM) we get a differential equation for $\tilde{x}$. To do so we must only keep in mind that all integrals are of the Ito type, i.e. the integrands are "delayed" with respect to dN_i and dw_i. We have

$$dxz^k = z^k dx + xd(z^k)$$

with $d(z^k)$ being expressible, using equation (MM), as linear combination of the dN_i. Substituting into this last equation from (MM) and (L') yields an equation of the form

$$d\tilde{x} = \tilde{A}\tilde{x}dt + \sum_{i=1}^{r} \tilde{B}_i\tilde{x}dw_i + \sum_{i=1}^{\delta} \tilde{C}_i\tilde{x}dN_i \tag{C}$$

where now $\tilde{A}$, $\tilde{B}_i$ and $\tilde{C}_i$ are constant. To complete the proof we simply let R be given by

$$R = \begin{bmatrix} 0 & & 0 \ldots 0 \\ 0 & & 0 \ldots 0 \\ \vdots & I_n & \ldots\ldots \\ 0 & & 0 \quad 0 \end{bmatrix}$$

The formula for the expectation follows from the fact that $\mathcal{E}f(x)(dN_i - \lambda_i dt) = \mathcal{E}f(x)dw_i = 0$ for smooth functions f.

4. MOMENTS AND COVARIANCES

In reference [4] it is shown how to calculate moments for linear differential equations with white noise coefficients. That is if

$$dx = Ax(t)dt + \sum_{k=1}^{\beta} C_k x dw_k$$

It is shown that the homogeneous of degree p monomials in $(x_1, x_2, \ldots, x_n)$, which we write as $x^{[p]}$, satisfy

$$\frac{d}{dt}\mathscr{E}x^{[p]} = (A - \sum_{k=1}^{\beta} \frac{1}{2} C_p^2)_{[p]} \; \mathscr{E}x^{[p]}$$

In [5] S. Marcus carried out a similar program for the moments associated with an equation of the form

$$dx = Axdt + \sum_{i=1}^{\delta} B_i x dN_i$$

Combining these results we obtain the following (compare with [6]):

Theorem 3: If $\tilde{x}$ satisfies equation (T) then $\mathscr{E}\tilde{x}^{[p]}$ satisfies

$$\frac{d}{dt}\mathscr{E}x^{[p]} = ((A - \frac{1}{2}\sum_{k=1}^{\beta} B_k^2)_{[p]} \mathscr{E}x^{[p]} + \sum_{i=1}^{\delta} (C_i + I)^{[p]} - I^{[p]}) \mathscr{E}x^{[p]} \tag{TH}$$

and $\mathscr{E}\tilde{x}(t)\tilde{x}'(\tau)$ is given for $\tau > t$ by

$$\mathscr{E}\tilde{x}^{[p]}(t)\tilde{x}^{[p]'}(\tau) = [\mathscr{E}\tilde{x}^{[p]}(t)\tilde{x}^{[p]'}(t)]e^{F_p(\tau - t)}$$

where F_p is the coefficient of $x^{[p]}$ on the right side of equation (TH).

Proof: From the moment equation is an easy application of the results of [4] and [5] coupled with the independence of the N_i and W_i processes. The equation for the covariance follows, for example, by differentiating $x^{[p]}(t)x^{[p]'}(\tau)$ with respect to τ, taking expectations and then solving the resulting ordinary differential equation.

In the special case of variance calculation it is perhaps more revealing to adopt the conventional matrix notation in place of $x^{[2]}$. That is, we let $\Sigma = \mathscr{E}x(t)x'(t)$ and note that from equation (T) we get

$$\dot{\Sigma} = \tilde{A}\Sigma + \Sigma\tilde{A}' + \sum_{i=1}^{\beta} B_i \Sigma B_i' + \sum_{i=1}^{\delta} \lambda_i C_i \Sigma C_i' - \lambda_i \Sigma$$

5. EXAMPLES

In this section we give a few applications of these ideas to illustrate some problems which can be routinely solved using theorems 1, 2, and 3.

Example 1: This is a well known example; the point is to do it neatly. We want to compute the covariance of a random telegraph wave which takes on the values ±1 and shifts from state 1 to state -1 according to the rule

$$\frac{d}{dt}\begin{bmatrix} p_1 \\ p_{-1} \end{bmatrix} = \begin{bmatrix} -\lambda & \lambda \\ \lambda & -\lambda \end{bmatrix}\begin{bmatrix} p_1 \\ p_{-1} \end{bmatrix}$$

We model the processes by letting $Z = \{-1, 1\}$. The differential equation (M) is then

$$dz(t) = -2z(t)dN(t); \quad z(0) = \pm 1$$

with N being a SCP with rate λ. It is clear that since $z^2(t) = 1$, $\mathscr{E}z(t)z(t) = 1$ and since

$$\frac{d}{dt}\,\mathcal{E}z(t) = -2\,\mathcal{E}\,z(t)$$

we have for $\tau > 0$, $\mathcal{E}z(t+\tau) = e^{-\lambda\tau}\,\mathcal{E}z(t)$ and hence

$$\mathcal{E}z(t)z(t+\tau) = e^{-2\lambda|\tau|}$$

A generalization is considered in [3].

Example 2: Consider a linear stochastic differential equation in $\mathbb{R}^n$

$$dx = A(t)xdt + bdw$$

where $A(t) = A\pm D$ is stochastic, taking on the values $A+D$ and $A-D$ and changing between them according to the transition law

$$\frac{d}{dt}\begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} -\lambda & \lambda \\ \lambda & -\lambda \end{bmatrix}\begin{bmatrix} p_1 \\ p_2 \end{bmatrix}$$

where p_1 is the probability of $(A+D)$ and p_2 is the probability of $A-D$. We want to calculate the covariance of the x process

$$\Sigma(t,\tau) = \mathcal{E}x(t)x'(\tau)$$

We arrive at a differential equation representation for the jump process by coding the states as 1 and -1. This gives, as in the last example,

$$dz = -2zdN$$

where N is an SPC of rate λ. Combining this with $dx = Axdt+zDxdt+bdw$ we get a system

$$d\begin{bmatrix} x \\ xz \end{bmatrix} = \begin{bmatrix} A & D \\ D & A \end{bmatrix}\begin{bmatrix} x \\ xz \end{bmatrix} dt + \begin{bmatrix} 1 \\ z \end{bmatrix} dw + \begin{bmatrix} 0 & 0 \\ 0 & -2 \end{bmatrix}\begin{bmatrix} x \\ xz \end{bmatrix} dN$$

We may take expectations using $\mathcal{E}zdw = 0$, $\mathcal{E}xz(dN-\lambda dt) = 0$, etc. to get

$$\mathcal{E}\begin{bmatrix} x(t) \\ x(t)z(t) \end{bmatrix} = \exp\begin{bmatrix} A & D \\ D & A-2I\lambda \end{bmatrix} t\,\mathcal{E}\begin{bmatrix} x(0) \\ zx(0) \end{bmatrix}$$

Using this formula we will be able to compute $\Sigma(t,\tau)$ from $\Sigma(t,t)$. Now for the variance equation. Introduce (recalling that $z^2 = 1$)

$$\Sigma(t,t) = \mathcal{E}\begin{bmatrix} x(t) \\ x(t)z(t) \end{bmatrix} [x'(t),\ x'(t)z(t)] = \begin{bmatrix} \Sigma_o(t,t) & \Sigma_1(t,t) \\ \Sigma_1(t,t) & \Sigma_o(t,t) \end{bmatrix}$$

The variance equation in these variables is given in terms of $\mathcal{E}z(t) = e^{-2\lambda t}\,\mathcal{E}\,z(0)$ by

$$\dot{\Sigma}(t,t) = \begin{bmatrix} A & D-I \\ D-I & A \end{bmatrix}\Sigma(t,t)+\Sigma(t,t)\begin{bmatrix} A & D-I \\ D-I & A \end{bmatrix}+\mathcal{E}\begin{bmatrix} bb' & z(t)bb' \\ z(t)bb' & bb' \end{bmatrix}$$

Then one sees rather easily that for $\tau > t$

$$\Sigma(t,t) = \mathcal{E}\begin{bmatrix} x(t) \\ z(t)x(t) \end{bmatrix} [x'(\tau)\ \ z(t)\ x'(\tau)] = \Sigma(t,t)\exp\begin{bmatrix} A & D \\ D & A-2I\lambda \end{bmatrix}(\tau-t)$$

It is perhaps, of some interest to note that $\Sigma_o(t,t)$ is, for $\Sigma_o(0,0) = 0$, positive definite for $t > 0$ exactly when the smallest subspace which contains b and is invariant under both A and D is $\mathbb{R}^n$. That is to say, when $\{A,D\}_{AA}b$ spans $\mathbb{R}^n$ where $\{A,D\}_{AA}$ indicates the associative algebra generated by A and D.

Example 3: We compute the variance of the solution of a linear system being driven by a generalized random telegraph wave. Suppose that in $\mathcal{R}^n$ we have

$$\dot{x} = Ax+bz(t)$$

where $z(t)$ is a finite state continuous time Markov process taking on values in $Z \subset R^1$ and satisfying both $p[z(t)] \equiv 0$ and equation (M). Ordinary calculus gives

$$\frac{d}{dt} xx' = Axx' + xx'A' + xb'z(t) + z(t)bx'$$

Thus to compute the variance we need an equation for $\mathcal{E}zx$. To get such an equation in convenient form we introduce $\tilde{z}$ as in section 2 and note that in terms of equation (MM) we have

$$dx\tilde{z}' = Ax\tilde{z}'dt + bz\tilde{z}'dt + \sum_{i=1}^{\delta} x\tilde{z}'M_i'dN_i$$

Consequently if λ_i is the rate of N_i

$$\frac{d}{dt} \mathcal{E}x\tilde{z}' = Ax\tilde{z}' + b\,\mathcal{E}z\tilde{z}' + \sum_{i=1}^{\delta} (\mathcal{E}x\tilde{z}')M_i'\lambda_i$$

This leaves us only to compute $\mathcal{E}z\tilde{z}$. This is accomplished using

$$\frac{d}{dt} \mathcal{E}\tilde{z}(t) = (\sum_{i=1}^{\delta} M_i\lambda_i)\mathcal{E}\tilde{z}(t)$$

and the relation $p(z) = 0$. Finally we see that these last two differential equations together with

$$\dot{\Sigma} = A\Sigma + \Sigma A + \mathcal{E}bx'z + \mathcal{E}xb'z$$

need to be solved to determine Σ. The solution of the equation for $\tilde{z}$ drives the $x\tilde{z}'$-equation whose solution, in turn, drives the Σ-equation.

REFERENCES

1. K. Ito, "On Stochastic Differential Equations," Memoirs of AMS, No. 4, 1951, Providence, R.I.
2. I.I. Gikhman and A.V. Skorokhod, Stochastic Differential Equations, Springer-Verlag, N.Y., 1972.
3. R.W. Brockett, "Stationary Covariance Generation Using Finite State Markov Processes," Proc. 1977 JACC, IEEE, 1977.
4. R.W. Brockett, "Lie Theory and Control Systems Defined on Spheres," SIAM J. on Applied Math., Vol. 25, No. 2, Sept. 1973, pp. 213-225.
5. S.I. Marcus, "Modeling and Analysis of Stochastic Differential Equations Driven by Point Processes," IEEE Trans. on Information Theory (to appear).
6. J. Morrison and J. McKenna, "Analysis of Some Stochastic Ordinary Differential Equation," in Stochastic Differential Equations, SIAM-AMS Proceedings, Vol. 6, 1973, pp. 97-161.

ON STOCHASTIC OPTIMAL CONTROL WITH AN EXPONENTIAL-OF-INTEGRAL COST CRITERION

P. R. KUMAR
Department of Mathematics
University of Maryland Baltimore County
5401 Wilkens Avenue
Baltimore, Maryland 21228

J. H. VAN SCHUPPEN
Department of Systems Science and Math
Washington University
Box 1040
St. Louis, Missouri 63130

Abstract

A theory of dynamic programming of partially observed systems is developed for situations where the cost criterion is the exponential of an integral form. A specific class of problems is solved.

1. INTRODUCTION

In deterministic optimal control the minimization of an integral cost criterion, $J = \int_0^T cdt$, is equivalent to the minimization of the criterion $J = m(\int_0^T cdt)$ whenever m is a monotone increasing function. However, in stochastic control this is not so, since the two criteria would then be $E[\int_0^T cdt]$ and $E[m(\int_0^T cdt)]$, and the minimizations of these criteria are not equivalent since

$$E[m(\int_0^T cdt)] \neq m[E \int_0^T cdt]$$

In this paper we consider those stochastic control problems where the cost is the exponential of an integral, $J = E[\exp(\int_0^T cdt)]$. Such a criterion has been considered earlier by Jacobson [1] and Speyer et al. [2] for the special case where the system is linear and the index of the exponent is a quadratic form.

We note that the integral cost commonly used possesses the property that at any time t the total cost may be additively decomposed as the sum of past and future costs,

$$\int_0^T cds = \int_0^t cds + \int_t^T cds.$$

Thus one minimizes the future cost-to-go and this leads to the concept of an optimal value function and the principle of optimality and accounts for the dynamic programming approach. Similarly the exponential of integral cost may be multiplicatively decomposed as the product of past and future costs,

$$\exp(\int_0^T cds) = \exp(\int_0^t cds) \quad \exp(\int_t^T cds).$$

In this paper we develop a theory of dynamic programming for this cost criterion. In Section 2 we show that a reasonable condition which allows dynamic programming is satisfied only by the above two cost criteria, thus excluding other monotone functions of an integral form.

The exponential of integral cost criterion may also be written as $\exp[\mu \int_0^T cds]$ with μ a constant. Then, it has a series representation

$$E[\exp(\mu \int_0^T cds)] = 1 + \mu E[\int_0^T cds] + \frac{\mu^2}{2} E[\int_0^T cds]^2 + \ldots.$$

Thus, as shown in [1] and [2], minimization of this cost entails minimization of some combination of all moments. Hence this criterion is useful when one is interested in minimizing not just the mean integral cost, but also, for example, its variance. Alteration of μ thus allows some shaping of the distribution of the integral loss. Also for small values of μ the first moment is dominant, and hence, in the limit as $\mu \to 0$, the exponential of integral cost includes the integral cost as a special case.

In some problems the exponential of integral cost problem may be converted to an integral cost problem by adjoining an additional variable to the system equations - and vice versa. However, a direct analysis of the exponential of integral cost criterion clarifies the separation of the cost into past and future components and yields intuitively appealing results.

In Section 2 we outline an abstract model and martingale optimality conditions due to Striebel [3] and in Section 3 we obtain optimality conditions in this model for the exponential of integral cost criterion. In Section 4 we impose, via the transformation of Wiener measure, the additional structure of a partially observed system of stochastic differential equations and in Section 5 we obtain necessary and sufficient dynamic programming conditions which parallel those of Davis and Varaiya [4] for the integral cost case. Finally, in Section 6, we present solutions to a class of systems introduced by Speyer et al. [2]. The results of this paper are from the thesis [5] of the first author, to which the reader is referred for more details.

2. ABSTRACT MODEL

We now present a model of Striebel [3] for stochastic control and a martingale characterization of optimality. Our presentation is brief and the reader is referred to the original source [3] for a fuller account. The discrete time version is presented in [6].

Definition 2.1: The stochastic control model

1) Given a set Ω, a time interval $[0,T]$ with $T<\infty$ and increasing σ-algebra families $\{F_t, t \varepsilon [0,T]\}$ and $\{D_t, t \varepsilon [0,T]\}$ satisfying $D_0 = \{\phi, \Omega\}$ and for all $t \varepsilon [0,T]$, $D_t \subset F_t$, let $L: \Omega \to R_+$ be F_T-measurable.

2) For every $t \varepsilon [0,T]$ let U_t be a set such that:

i) U_0 is a singleton,

ii) For every $\xi_t \varepsilon U_t$ there exists a probability measure P_{ξ_t} on on (Ω, F_t),

iii) $E_{\xi_T}[L] < \infty$ for all $\xi_T \varepsilon U_T$.

iv) $P_{\pi_{t,T}\xi_T}(A) = P_{\xi_T}(A)$ for all $A \varepsilon F_t$ where for every $0 \leq s \leq t \leq T$, $\pi_{s,t}: U_t \xrightarrow{\text{onto}} U_s$ satisfies a) $\pi_{t,t}\xi_t = \xi_t$

b) $\pi_{s,\tau}\pi_{\tau,t}\xi_t = \pi_{s,t}\xi_t$ for $\tau \varepsilon [s,t]$

Interpretation. We interpret $\{F_t, t \varepsilon [0,T]\}$ as the family to which all processes are adapted and D_t as the information available to the controller at time t. We interpret U_t as the set of admissible control laws upto t, and account for the influence of control laws on events by associating each control law upto time t, ξ_t with a probability measure P_{ξ_t} on the σ-algebra F_t of events upto time t. $\pi_{s,t}$ is a mapping which truncates a control law at time s. Condition 2.iv is merely a causality requirement that the probability of an event $A \varepsilon F_t$, which occurs before time t, be determined only by the portion of the control law upto time t. We interpret L as the loss accruing to the controller.

We shall denote conditional expectation w.r.t a σ-algebra D under measure ξ by $E^D_\xi(\cdot)$ and shall denote $E^D_\xi(I_A)$ by $P^D_\xi(A)$. The following is the requirement of optimality.

Definition 2.2: Conditional optimality

$\xi_T \varepsilon U_T$ is said to be conditionally optimal if, for each $t\varepsilon[0,T]$,

$$E^{D_t}_{\xi^o_T}(L) \le E^{D_t}_{\xi_T}(L) \text{ a.s. } P^{D_t}_{\pi_{t,T}\xi^o_T} \qquad \forall\xi_T \text{ with } \pi_{t,T}\xi_T = \pi_{t,T}\xi^o_T .$$

This is interpreted as saying that if the controller has played ξ^o_T upto t, then with the knowledge possessed at t, the controller can do no better than play ξ^o_T the rest of the way - a reasonable requirement. This includes (consider t=0 in above) the traditional definition of optimality.

Definition 2.3: The ε-lattice property

The model is said to possess the ε-lattice property if, for $0\le t\le T$ and any $\xi_1,\xi_2\varepsilon U_T$ such that $\pi_{t,T}\xi_1 = \pi_{t,T}\xi_2$, and $\varepsilon>0$, there exists $\xi_0\varepsilon U_T$ such that $\pi_{t,T}\xi_0 = \pi_{t,T}\xi_i$, i = 1,2 and

$$E^{D_t}_{\xi_o}(L) \le E^{D_t}_{\xi_i}(L) + \varepsilon \qquad \text{for } i = 1,2.$$

This definition is just a statement about the richness of the class of admissible strategies.

Definition 2.4: The conditional loss function

The conditional loss function is defined for $0\le t\le T$ and $\xi_t\varepsilon U_t$ by

$$F_t(\omega,\xi_t) = P^{D_t}_{\xi_t} - \underset{\{\xi'_T\varepsilon U_T|\pi_{t,T}\xi'_T = \xi_t\}}{\text{essential infimum}} E^{D_t}_{\xi'_T}(L)$$

Theorem 2.5: Necessary and sufficient martingale optimality conditions

If there exists a nonnegative function $F_t(\omega,\xi_t)$ adapted to $\{D_t,t\varepsilon[0,T]\}$ and such that

i) $F_T(\omega,\xi_T) = E^{D_T}_{\xi_T}(L)$ a.s. $P^{D_T}_{\xi_T}$ for all $\xi_T\varepsilon U_T$

ii) For every $\xi_T\varepsilon U_T$, $F_t(\omega,\pi_{t,T}\xi_T)$ is a submartingale on $\{D_t,t\varepsilon[0,T]\}$ under P_{ξ_T}, and a martingale for ξ^o_T,

then ξ^o_T is conditionally optimal. Moreover the above conditions are necessarily satisfied by the conditional loss function $\hat{F}_t(\omega,\pi_{t,T}\xi_T)$ whenever the system possesses the ε-lattice property.

Proof. See Theorem 2 of [3].

Theorem 2.6: Stochastic dynamic programming

If the system possesses the ε-lattice property, the conditional loss function $\hat{F}_t(\omega,\xi_t)$ is the unique nonnegative function adapted to $\{D_t,t\varepsilon[0,T]\}$ satisfying

$$F_s(\omega,\xi_s) = P^{D_s}_{\xi_s} - \underset{\{\xi'_t\varepsilon U_t|\pi_{s,t}\xi'_t = \xi_s\}}{\text{essential infimum}} E^{D_s}_{\xi'_t}[F_t(\omega,\xi'_t)] \tag{1}$$

with boundary condition given by condition (i) on Theorem 2.5.

Proof. See Theorem 5 of [3].
We call (1) the stochastic dynamic programming equation.

3. EXPONENTIAL-OF-INTEGRAL COST CRITERION

We now do away with the generality of the loss function of Section 2 and assume that the controllers evaluation of the loss, which is all that really matters, is the exponential of an integral form. We assume that,

$$E_{\xi_T}^{D_T}(L) = \exp[\int_0^T c_t(\omega,\pi_{t,T}\xi_T)dt] \quad \forall \xi_T \varepsilon U_t$$

where $c_t(\omega,\pi_{t,T}\xi_T)$ is nonnegative, jointly measurable and adapted to $\{D_t, t\varepsilon[0,T]\}$. Our first result is the following:

Theorem 3.1. Let $F_t(\omega,\xi_t)$ and $V_t(\omega,\xi_t)$ be two nonnegative, integrable functions both adapted to $\{D_t, t\varepsilon[0,T]\}$ and related by

$$F_t(\omega,\xi_t) = V_t(\omega,\xi_t)\exp[\int_0^T c_\tau(\omega,\pi_{\tau,t}\xi_t)d\tau] .$$

Then $F_t(\omega,\xi_t)$ satisfies the stochastic dynamic programming equation (1) if and only if for every $0\leq s\leq t$, $V_t(\omega,\xi_t)$ satisfies

$$V_s(\omega,\xi_s) = P_{\xi_s}^{D_s} - \underset{\{\xi_t'\varepsilon U_t | \pi_{s,t}\xi_t' = \xi_s\}}{\text{essential infimum}} E_{\xi_t'}^{D_s}[V_t(\omega,\xi_t')\exp(\int_s^t c_\tau(\omega,\pi_{\tau,t}\xi_t')\, d\tau]$$

Proof. See Theorem 2.9 of [5].

We have invoked here the multiplicative decomposition of this cost criterion, in contrast to the usual invocation of the additive decomposition of integral costs. Our results are therefore the analogs of existing results for integral costs, see Theorem 6 of [3], Theorem 3.1 of [4] or Theorem 4 of [7]. The theorem above is the analog of the principle of optimality and $V_t(\omega,\xi_t)$ that of the optimal value function, and henceforth we shall refer to them as such.

A question that naturally arises at this point is whether there exists any other cost criterion which combines, and therefore generalizes the properties of additive and multiplicative decomposability. That is, what are all the functions $m:R \to R$ (not necessarily monotone) such that a cost criterion $m(\int_0^T cds)$ possesses the property that there exist functions h and g such that for each t,

$$m(\int_0^T cds) = h(\int_0^t cds) \quad m(\int_t^T cds) + q(\int_0^t cds) \tag{2}$$

Note that if m satisfies (2) with a nonnegative valued h, a theory of dynamic programming can be developed for the criterion $m(\int_0^T cds)$. The integral cost satisfies (2) with $h\equiv 1$ and $g\equiv m\equiv$the identity function, while the exponential of integral cost satisfies the above with $g\equiv 0$ and $h\equiv m\equiv$ exponential function. We now show that these are the only interesting cases to satisfy (2).

Lemma 3.2. $m:R \to R$ satisfies the functional equation

$$m(p+f) = h(p)m(f) + g(p) \qquad \forall p,f\varepsilon R \tag{3}$$

with some $h,g:R \to R$, both differentiable at 0, if and only if m is one of the following functions:

$$\begin{aligned} &\text{i)} \quad m(x) = c \\ &\text{ii)} \quad m(x) = a+bx \qquad\qquad (4) \\ &\text{iii)} \quad m(x) = ae^{bx}+c \end{aligned}$$

Proof. From (3), we have that $m(x) = h(0)m(x)+g(0)$. If $h(0)\neq 1$, then we solve and obtain (4.i). If instead $h(0)=1$, then $g(0)=0$ and from (3), we have

$$\frac{m(x+\varepsilon)-m(x)}{\varepsilon} = m(x)\,\frac{[h(\varepsilon)-1]}{\varepsilon} + \frac{g(\varepsilon)}{\varepsilon}$$

Since $\dot{h}(0)$ and $\dot{g}(0)$ exist, $\dot{m}(x)$ also exists and

$$\dot{m}(x) = \dot{h}(0)m(x)+\dot{g}(0)$$

If $\dot{h}(0)=0$, then (4.ii) is satisfied. If $\dot{h}(0)\neq 0$ then (4.iii) is valid. The converse of the lemma is verified by substitution.

Now, if we adopt $m(\int_0^T c\,ds)$ as a cost criterion with m given by (4.i), then we have a constant cost which is of no practical use, while (4.ii) and (4.iii) are just scalings and/or translations of the integral and exponential cost criteria and thus included in the two. Thus these are the only two useful criteria.

4. PARTIALLY OBSERVED STOCHASTIC DIFFERENTIAL EQUATIONS MODEL

In Section 3 we have specialized to the situation where the cost criterion is the exponential of an integral, while retaining the generality of Striebel's model. In this section we dispense with this remaining generality and impose on the model the structure obtained from a partially observed system of stochastic differential equations with Brownian motion disturbances.

To be consistent with the general model which identifies a control law with a probability measure and which also requires the invariance of observation σ-algebras with control actions, we adopt the Girsanov definition of the solution of stochastic differential equations. Such an approach has the added advantage that the usual Lipschitz conditions imposed in connection with stochastic differential equations are not necessary. For brevity, and also because this method has been adopted earlier by Benes [8], Duncan and Varaiya [9] and Davis and Varaiya [4] we present only an outline of the procedure. The reader is referred to [5] for details.

We assume the following:

i) Ω is the space of R^{n+m}-valued continuous functions on $[0,T]$.

ii) For each $t\varepsilon[0,T]$, F_t is the σ-algebra on Ω generated by cylinder sets of the form $\{z\varepsilon R:z(s)\varepsilon A\}$ where $0\leq s\leq t$ and A is a Borel set in R^{n+m}.

iii) For each t, D_t is a similar σ-algebra generated by the last m-components of $z\varepsilon\Omega$.

iv) σ is an $(n+m)x(n+m)$ matrix valued function such that $\sigma_{ij}:[0,T]x\Omega\to R$ and $\sigma_{ij}(t,\cdot)$ is F_t-measurable.

v) $\underline{U}\subset R^r$ is a Borel set.

vi) $g:[0,T] \times \Omega \times U\to R^{n+m}$ is such that $g(t,\cdot,u)$ is F_t-measurable.

vii) For each $\xi_t\varepsilon U_t$, $u^{\xi_t}:[0,t] \times \Omega \to \underline{U}$ is such that $u(\tau,\cdot)$ is D_τ-measurable.

viii) P_o is the Wiener measure on (Ω, F_T) with $\{w^*_t, t\varepsilon[0,T]\}$ the corresponding Wiener process.

ix) $\{z_t, t\varepsilon[0,T]\}$ is sample continuous satisfying the stochastic differential equation

$$z_t = z_0 + \int_0^t \sigma(s,z)dw^*_s.$$

x) For $0\leq s\leq t$ and $\xi_t \varepsilon U_t$, we define

$$\rho_s^t(u^{\xi_t}) = \exp(\int_s^t [\sigma^{-1}(\tau,z)g(\tau,z,u^{\xi_t}(\tau,z)]'dw^*_\tau$$
$$- \frac{1}{2} \int_s^t |\sigma^{-1}(\tau,z)g(\tau,z,u^{\xi_t}(\tau,z))|^2 d\tau)$$

xi) Assume $E_0^{F_s}[\rho_s^t(u^{\xi_t})] = 1$ a.s.P_0 (where $E_0(\cdot)$ denotes expectation under measure P_0.

Under the above assumptions and other conditions of a technical nature listed in [5] it follows from the work of Girsanov [10] that P_{ξ_t} defined by

$$P_{\xi_t}(A) = \int_A \rho_o^t(u^{\xi_t})dP_0 \qquad \text{for all } A\varepsilon F_t$$

is a probability measure on (Ω, F_t), and furthermore under this probability measure $\{z_s, s\varepsilon[0,t]\}$ satisfies,

$$z_s = z_0 + \int_0^s g(\tau,z,u^{\xi_t}(\tau,z))ds + \int_0^s \sigma(\tau,z)dw_\tau \tag{5}$$

where $\{w_\tau, \tau\varepsilon[0,T]\}$ defined by

$$dw_s = dw^*_s - \sigma^{-1}(s,z)g(s,z,u^{\xi_t}(s,z))ds$$

is a Brownian motion under P_{ξ_t}.

We thus achieve our goal of imposing the structure of (5), a partially observed system of stochastic differential equations, on the model. We mention that condition (xi) is non-trivial, and various sufficiency conditions may be found in [8] and [9]. Also in (5) we note that g and σ are allowed to depend on the whole past of z, and not just on its instantaneous present value. The reader is referred to [5] for a verification that this model satisfies the requirements of Section II. We also assume that the cost criterion is such that

$$E_\xi^{D_T}(L) = \exp[\int_0^T c(t,z,u^{\xi_T}(t,z))dt] \qquad \text{for all } \xi_T\varepsilon U_T$$

where $c: [0,T] \times \Omega \times \underline{U}\rightarrow[0,K]$ is jointly measurable and $c(t, ,u)$ is D_t-measurable.

As seen in Section 2, the ε-lattice property is very useful since it establishes many other results. The following result is therefore very important.

Lemma 4.1: The ε-lattice property for the s.d.e. model

The stochastic differential equations model above possesses the ε-lattice property.

Proof. For ξ_1 and ξ_2 of Definition 2.3, define

$$S = \{\omega\varepsilon\Omega: E_{\xi_2}^{D_t}(L)(\omega) - E_{\xi_1}^{D_t}(L)(\omega) \leq 0\}$$

It is then possible to show that ξ_0 associated with

$$u^{\xi_0}(t,z) = \begin{cases} u^{\xi_1}(t,z) & \text{for } z\varepsilon S \\ u^{\xi_2}(t,z) & \text{for } z\varepsilon S^c \end{cases}$$

satisfies the requirements of Definition 2.3.

Previous results [4] in dynamic programming have utilized a property of "relative completeness" defined by Rishel [7]. However the proof of relative completeness for a stochastic differential equations model involves a lengthy Zorn's lemma argument. Here, utilizing the results of Striebel [3], the weaker ε-lattice property is sufficient for our purposes and as shown above is easily proven.

5. DYNAMIC PROGRAMMING CONDITIONS

In this section we obtain necessary and sufficient dynamic programming conditions for the partially observed system of stochastic differential equations with an exponential of integral performance criterion. First we need the following definitions which parallel those for integral costs.

Definition 5.1: The value function

$V_t(\omega,\xi_t)$ defined by

$$\hat{F}_t(\omega,\xi_t) = V_t(\omega,\xi_t)\exp[\int_0^t c(\tau,z,u^{\xi_t}(\tau,z))d\tau]$$

where $\hat{F}_t(w,\xi_t)$ is the conditional loss function, will be called the value function.

Definition 5.2: Value decreasing control law

A control law ξ_T is said to be value decreasing if for every $0\leq s\leq t\leq T$, we have $E_{\xi_T}^{D_s}[V_t^{\xi_T}] \leq V_s^{\xi_T}$ a.s. $P_{\xi_T}^{D_s}$

The following lemma then follows.

Lemma 5.3

If $\xi_T^o\varepsilon U_T$ is conditionally optimal, then it is value decreasing.

Proof. See Lemma 2.13 of [5].

From here on we write $z' = (x':y')$ and identify y with the observed components of z. Correspondingly we partition g as $g' = (g_1',g_2')$. We also denote $E_\xi^{D_t}(g_2)$ by $\hat{g}_2^{\xi_t}$. The following gives dynamic programming conditions.

Theorem 5.4: Necessary and sufficient dynamic programming conditions

$\xi^o\varepsilon U_T$ is optimal, if and only if, there exists $J^o\varepsilon R$ and for each value decreasing $\xi\varepsilon U_T$ there exists a real valued process $\{a_t^\xi\}$ and an R^m valued $\{b_t^\xi\}$, both jointly measurable and adapted to $\{D_t, t\varepsilon[0,T]\}$ and satisfying,

i) $\int_0^T |b_t^\xi|^2 dt<\infty$ a.s.

ii) $E_o \int_0^t b_s^\xi dy_s = 0$ for all $t\varepsilon[0,T]$.

and such that V_t^ξ defined by

$$V_t^\xi = J^o + \int_0^t a_s^\xi ds + \int_0^t b_s^\xi dy_s$$

is nonnegative and satisfies,

iii) $V_T^{\xi} = 1$ a.s.

iv) $a_t^{\xi} + b_t^{\xi}\, \hat{g}_{2t}^{\xi} + V_t^{\xi} C_t^{\xi} \geq 0 = a_t^{\xi^o} + b_t^{\xi^o}\, \hat{g}_{2t}^{\xi^o} + V_t^{\xi^o} C_t^{\xi^o}$ a.s. in $[0,T] \times \Omega$.

Furthermore J^o is the optimal cost of the system.

Outline of proof

The sufficiency of the optimality conditions is easily proven, see Theorem 2.14 of [5]. For necessity, using Lemma 5.3 we show that V_t^{ξ} is a right continuous supermartingale. Then we show that γ_t^{ξ} defined by $\gamma_t^{\xi} = \ln V_t^{\xi}$ is a potential of class (D) and we apply the supermartingale decomposition theorem of Meyer [11]. We then define $\beta_t^{h,\xi} = \frac{1}{h} [\gamma_t - E_{\xi}^{D_t}(\gamma_{t+h}^{\xi})]$ and show that $\{\beta_t^{h,\xi} : h>0\}$ is a uniformly integrable collection. From this, and following the arguments in [4], which utilize a martingale representation theorem, we arrive at an integral representation of γ_t^{ξ}. The final result is then obtained by use of the differentiation rule and some limiting arguments.

In this paper, we do not treat the special cases of completely observed systems and completely observed Markov systems, even though we believe that such extensions may be carried out much as in [4].

6. LEG SYSTEM: NO PLANT NOISE CASE

In this section we solve a specific problem first suggested by Speyer et al. [2]. Consider the linear system,

$$dx(t) = F(t)x(t)dt + G(t)u(t)dt$$

$$dy(t) = h(t)x(t)dt + V^{\frac{1}{2}}(t)dv(t)$$

where $x(0)\ N(\bar{x}_0, P_0)$ and $y(0) = 0$, $x(t)$ is the state and $y(t)$ the observations, $v(\cdot)$ is a standard Brownian motion. The cost criterion is

$$J(u) = E[\mu \exp \frac{\mu}{2} \{x'(T)Q_f x(T) + \int_0^T (u'(s)R(s)u(s) + x'(s)Q(s)x(s))ds\}]$$

This system may be modeled by a measure transformation method, in spite of the degeneracy of state noise, and the details are given in [5]. The optimal control given by Speyer et al. contains some slight mistakes. We give here the correct optimal control and present a martingale proof.

Theorem. For the above problem, under the condition that $P^{-1}(T) > \mu Y$ where P and Y are defined below, the optimal control is given by

$$u^o(t) = -R^{-1}(t)G'(t)[L(t)x(t) + M(t)\ (t)]$$

where

$$dx(t) = F(t)x(t)dt + G(t)u(t)dt + K(t)[dy(t) - H(t)x(t)dt];\ x(0) = \bar{x}_0$$

$$\dot{\eta}(t) = [\int_0^t \Phi'(s,T)Q(s)\Phi(s,T)ds]\Phi(T,t)G(t)u(t);\ \eta(0) = 0$$

and $L(t) = C_1(t) + \Phi'(T,t)[\int_0^t \Phi'(s,T)Q(s)\Phi(s,T)ds]C_2'(t)$

$$M(t) = C_2(t) + \Phi'(T,t) + \Phi'(T,t)[\int_0^t \Phi'(s,T)Q(s)\Phi(s,T)ds]C_4(t)$$

$$\dot{\Phi}(t,s) = F(t)\Phi(t,s);\ \Phi(s,s) = I$$

$$\dot{C}_1(t) = -F'(t)C_1(t) - C_1(t)F(t) - \mu C_1(t)K'(t)V(t)K(t)C_1(t)$$
$$+ L'(t)G(t)R^{-1}(t)G'(t)L(t);\ C_1(T) = Y[I - \mu P(T)Y]^{-1}$$

$$\dot{C}_2(t) = -F'(t)C_2(t) + L'(t)G(t)R^{-1}(t)G'(t)M(t);\ C_2(T) = -[I - \mu YP(T)]^{-Y}$$
$$\dot{C}_4(t) = M'(t)B(t)R^{-1}(t)B'(t)M(t);\ C_4(T) = \mu[P^{-1}(T) - \mu Y]^{-1}$$
$$Y = Q_f + \int_0^T \Phi'(t,T)Q(t)\Phi(t,T)dt$$
$$K(t) = P(t)H'(t)V^{-1}(t)$$
$$P(t) = F(t)P(t) + P(t)F'(t) - P(t)H'(t)V^{-1}(t)H(t)P(t);\ P(0) = P_0$$

Proof. Define

$$F_t^u = \mu\alpha(t)\left(\exp \frac{\mu}{2} \begin{pmatrix} x(t) \\ \eta(t) \end{pmatrix}' \begin{pmatrix} C_1(t) & C_2(t) \\ C_2'(t) & C_4(t) \end{pmatrix} \begin{pmatrix} x(t) \\ \eta(t) \end{pmatrix} + \int_0^t \begin{pmatrix} u(s) \\ \eta(s) \end{pmatrix}' \begin{pmatrix} R(s) & G'(s)\Phi'(T,s) \\ \Phi(T,s)G(s) & 0 \end{pmatrix} \begin{pmatrix} u(s) \\ \eta(s) \end{pmatrix} ds\right)$$

where $\dot{\alpha}(t) = -\frac{1}{2}\mu\alpha(t)$ trace $[V(t)K'(t)C_1(t)K(t)]$; $\alpha(T) = [\det(I- YP(T))]^{-\frac{1}{2}}$

Then, using $x(s) = \Phi(s,T)x(T) - \int_0^T \Phi(s,\tau)G(\tau)u(\tau)d\tau$, and the fact that $x(T)$ is conditionally normally distributed with mean $x(T)$ and covariance $P(T)$, we obtain

$$F_T^u = E^{D_T}[\mu\exp \frac{\mu}{2}\ x'(T)Q_f x(T) + \int_0^T (u'(s)R(s)u(s) + x'(s)Q(s)x(s))ds\ ]$$

Also using the differentiation rule, we obtain

$$dF_t^u = \exp\left(\frac{\mu}{2}\begin{pmatrix} \hat{x} \\ \eta \end{pmatrix}' \begin{pmatrix} C_1 & C_2 \\ C_2' & C_4 \end{pmatrix} \begin{pmatrix} \hat{x} \\ \eta \end{pmatrix} + \int_0^T \begin{pmatrix} u \\ \eta \end{pmatrix}' \begin{pmatrix} R & G'\Phi'(T,s) \\ \Phi'(T,s)G & 0 \end{pmatrix} \begin{pmatrix} u \\ \eta \end{pmatrix} ds\right).$$
$$\frac{1}{2}\mu^2\alpha(t)[u + R^{-1}G'(Lx+M\eta)]'\ R[u + R^{-1}G'(Lx+M\eta)]\ dt + dm(t)$$

where $m(t)$ is a martingale under P_u with respect to $\{D_t, t\varepsilon[0,T]\}$. The optimality of u^o then follows from Theorem 2.5.

We mention here that the optimality of the controls for the other continuous time LEG systems may also be rigorously proved by martingale methods. The details are given in [5].

7. CONCLUSIONS

In this paper we have developed a theory of dynamic programming for the exponential of integral performance criterion, which is the parallel of the theory for integral costs.

The theory would possibly be more complete if the cases of completely observed systems and completely observed Markov systems were also analyzed. We believe such extensions are possible along the lines of [4], and could be the topic of a future paper. Also some technical conditions, such as the requirement of uniform boundedness of c, and the restriction of our attention to value decreasing controls could perhaps also be dispensed with. Some work in the latter context has been done by Elliott [12] for systems with integral costs. Another potential area of research is the extension of the results of this paper to jump processes. Such an extension, for the case of integral costs has been done by Boel and Varaiya [13].

8. REFERENCES

1) D. H. Jacobson, "Optimal stochastic linear systems with exponential criteria and their relation to deterministic differential games," IEEE Transactions on Automatic Control, Vol. AC-18, 2, pp. 124-131, 1973.
2) J. L. Speyer, J. Deyst and D. H. Jacobson, "Optimization of stochastic linear systems with additive measurement and process noise using exponential performance criteria," IEEE Transactions on Automatic Control, Vol. AC-19, 4, pp. 358-366, 1974.
3) C. Striebel, "Martingale optimality conditions for the optimal control of continuous time stochastic systems," Memorandum, University of Minnesota.
4) M.H.A. Davis and P. Varaiya, "Dynamic programming conditions for partially observable stochastic systems," SIAM Journal on Control, Vol. 11, 2, pp. 226-261, 1973.
5) P. R. Kumar, "Stochastic optimal control and stochastic differential games," D.Sc. Thesis, Washington University, 1977.
6) C. Striebel, "Optimal control of discrete time stochastic systems," Lecture Notes in Economics and Mathematical Systems, Vol. 110, Springer-Verlag, Berlin, 1975.
7) R. Rishel, "Necessary and sufficient dynamic programming conditions for continuous time stochastic optimal control," SIAM Journal on Control, Vol. 8, 4, pp. 559-571, 1970.
8) V. E. Benes, "Existence of stochastic control laws," SIAM Journal on Control, Vol. 9, 3, pp. 446-472, 1971.
9) T. Duncan and P. Varaiya, "On the solutions of a stochastic control system," SIAM Journal on Control, Vol. 9, 3, pp. 354-371, 1971.
10) I. V. Girsanov, "On transforming a certain class of stochastic processes by absolutely continuous substitution of measures," Theory of probability and its applications, Vol. 5, pp. 285-301, 1960.
11) P. A. Meyer, "Probability and potentials," Blaisdell, Waltham, Massachusetts, 1966.
12) R. J. Elliott, "The optimal control of a stochastic system," Memorandum, University of Hull, England.
13) R. K. Boel and P. Varaiya, "Optimal control of jump processes," SIAM Journal on Control and Optimization, Vol. 15, 1, pp. 92-119, 1977.

ON THE STATE-SPACE APPROACH IN MODELING DATA-COMMUNICATION NETWORKS*

JAMES S. MEDITCH
Department of Electrical Engineering
University of Washington
Seattle, Washington 98195

ABSTRACT

The state-space approach for modeling data flow in store-and-forward digital data-communication networks is presented and analyzed. The roles that controllability and observability play in data routing are developed, and it is shown how multivariable control theory can be applied to yield feedback routing policies which regulate queue lengths at all nodes throughout the network in the presence of heavy input traffic. The study also examines the implications of delayed information patterns.

1. INTRODUCTION

Recent research [1-5] has shown that the use of state-space methods offers considerable advantage and flexibility over the traditional queueing theory formulation [6,7] in the modeling of data flow in store-and-forward data-communication networks. This appears to be particularly relevant in the case of the routing problem which consists of specifying rules which define paths for the efficient transmission of information through the network. In this problem, efficiency is commonly expressed in terms of average message time delay or by queue lengths at nodes.

In this paper, we examine the state-space approach from the viewpoint of regulating queue lengths at all nodes throughout the network in the presence of heavy input traffic. We present the network flow state-space model in Section 2 where we also give necessary and sufficient conditions for data flow controllability and observability. In Section 3, we apply multivariable control theory to develop feedback routing policies, and show the effect of delayed information patterns in feedback routing. We close in Section 4 with a discussion of our results.

2. MODEL FORMULATION AND ANALYSIS

Consider a data network which has N nodes that are indexed by the set $\mathcal{N} = \{1, 2, \ldots, N\}$. Let (i, j), i, j $\in \mathcal{N}$, denote a directed link from node i to node j (see Fig. 1), and define $\mathcal{L}$ to be the set of all such links that actually exist in the network. With respect to $\mathcal{L}$, we shall need for each $i \in \mathcal{N}$, the two index sets

$$P(i) = \{j \in \mathcal{N}: (i,j) \in \mathcal{L}\}$$

and

$$Q(i) = \{\ell \in \mathcal{N}: (\ell,i) \in \mathcal{L}\} .$$

---

* Research supported in part by the National Science Foundation, Washington, D.C., under Grant No. ENG73-03729 A02.

We let t = 0, 1, ... denote discrete time with an assumed sampling period T > 0 that is not necessarily uniform for all time.

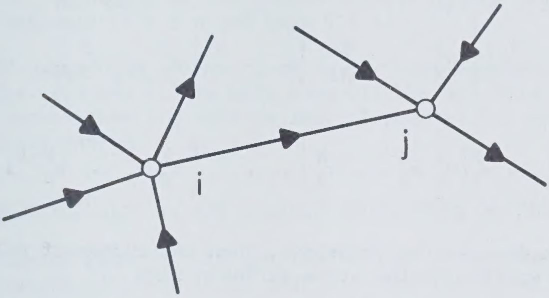


Fig. 1. Network node and link structure.

We select the bit to be the unit of data with which we shall deal, although our analysis in the sequel is equally valid for bytes, messages, packets, etc. With this in mind, we define the basic network variables

$x_i^k(t)$ = number of bits at node i at time t whose destination is node k where $i, k \in N$ and $k \neq i$

$u_{ij}^k(t)$ = rate (in bits/sec) at which data whose destination is node k are transmitted over link (i,j) during the interval [t, t+1) where $i, j, k \in N$, $k \neq i$, and $(i,j) \in L$

$w_i^k(t)$ = number of bits that arrive at node i from outside the network during the interval [t, t+1) and whose destination is node k where $i, k \in N$ and $k \neq i$.

From conservation of data flow, it follows that

$$x_i^k(t+1) = x_i^k(t) - T \sum_{j \in P(i)} u_{ij}^k(t) + T \sum_{\substack{\ell \in Q(i) \\ \ell \neq k}} u_{\ell i}^k(t) + w_i^k(t) \tag{1}$$

where t = 0, 1, ... ; $i, k \in N$, and $k \neq i$.

From a control theoretic standpoint, we view the x_i^k, u_{ij}^k and w_i^k as state, control, and command (input) variables, respectively. If we thus define

$$x = [x_1^2 \; x_1^3 \; \ldots \; x_1^N \mid x_2^1 \; x_2^3 \; \ldots \; x_2^N \mid \ldots \mid x_N^1 \; x_N^2 \; \ldots \; x_N^{N-1}]'$$

$$\begin{aligned} u = [u_{12}^2 \; u_{12}^3 \; \ldots \; u_{12}^N \mid u_{13}^2 \; u_{13}^3 \; \ldots \; u_{13}^N \mid \; . \; . \; . \; \mid u_{1N}^2 \; u_{1N}^3 \; \ldots \; u_{1N}^N \mid \\ u_{21}^1 \; u_{21}^3 \; \ldots \; u_{21}^N \mid \; . \qquad . \qquad . \; \mid u_{2N}^1 \; u_{2N}^3 \; \ldots \; u_{2N}^N \mid \\ . \; . \; . \mid u_{N1}^1 \; u_{N1}^2 \; \ldots \; u_{N1}^{N-1} \mid . \; . \; . \mid u_{N,N-1}^1 \; u_{N,N-1}^2 \; \ldots \; u_{N,N-1}^{N-1}]' \end{aligned} \tag{2}$$

and

$$w = [w_1^2 \; w_1^3 \; \ldots \; w_1^N \mid w_2^1 \; w_2^3 \; \ldots \; w_2^N \mid . \; . \; . \mid w_N^1 \; w_N^2 \; \ldots \; w_N^{N-1}]'$$

where the prime denotes the transpose, then the system of relations in Eq. (1) can be written in the state variable form

$$x(t+1) = x(t) + Bu(t) + Gw(t) \tag{3}$$

where $t = 0, 1, \ldots$. We take x to be an n-vector, u a p-vector, and w an r-vector. We observe that B is $n \times p$ and has only three types of entries: 0, $+T$ and $-T$; while G is $n \times r$ and is comprised of zeros and ones.

We observe that $w(t)$ models the arrival of data to the network from external sources over $[t, t+1)$, while the routing policy $u(t)$ that is in effect over the same interval is initiated at time t and acts independently of the exact value of $w(t)$. This formulation is realistic since we cannot expect, in general, to know at time t what the input traffic over $[t, t+1)$ will be. Indeed, if $w(t)$ is known a priori, then routing policies that partially or totally cancel its effect can be determined. On the other hand, the model clearly permits the description of $w(t)$ as a random process or as a sequence of known form, e.g., a polynomial with unknown coefficients, but otherwise arbitrary.

To prevent data transmission throughout the network from being blocked in the presence of heavy input traffic $w(t)$, we choose to regulate the queue lengths at all nodes in the network for such input traffic. To this end, we assume a desired queue vector $\bar{x}$ = constant n-vector which is consistent with nodal storage capacity, and define the queueing error to be

$$e(t) = x(t) - \bar{x} \tag{4}$$

From Eqs. (3) and (4), we see immediately that

$$e(t+1) = e(t) + Bu(t) + Gw(t) \tag{5}$$

where $t = 0, 1, \ldots$. Equation (5) describes the reformulated model in which we now view $w(t)$ as a disturbance whose effect we seek to nullify via $u(t)$.

Two very important properties of the model are immediate. First, we see that the system in Eq. (5) is controllable if and only if $p \geq n$ and rank $B = n$. Second, if we take the measured output to be

$$y(t) = Ce(t) \tag{6}$$

where C is m x n, we note that the system defined by Eqs. (5) and (6) is observable if and only if $m \geq n$ and rank $C = n$.

For reasons that will become clear in the next section, we assume that the model in Eqs. (5) and (6) is both controllable and observable. In addition, for convenience and with no loss of generality, we take C to be the n x n identity matrix.

3. ROUTING POLICIES AND DELAYED INFORMATION PATTERNS

In our study of routing, the problem is that of regulating e(t) to zero in the presence of heavy input traffic. The latter we take to be of the form shown below in Fig. 2 relative to the nominal input traffic $\bar{w}$. We assume that the amplitudes of the jumps are bounded, but otherwise arbitrary, and that the occurrence times of these jumps are "on the average" greater than 2nT time units apart. The relevance of this latter assumption will become apparent later. For now it suffices to note that nT can be made small via the choice of T. Given this information and the model in Eq. (5) in which we assume that rank $B = n$, we wish to determine u(t) such that e(t) is driven to zero and maintained there. Utilizing the notion of "system type" from multivariable control theory [8], we append to Eq. (5)

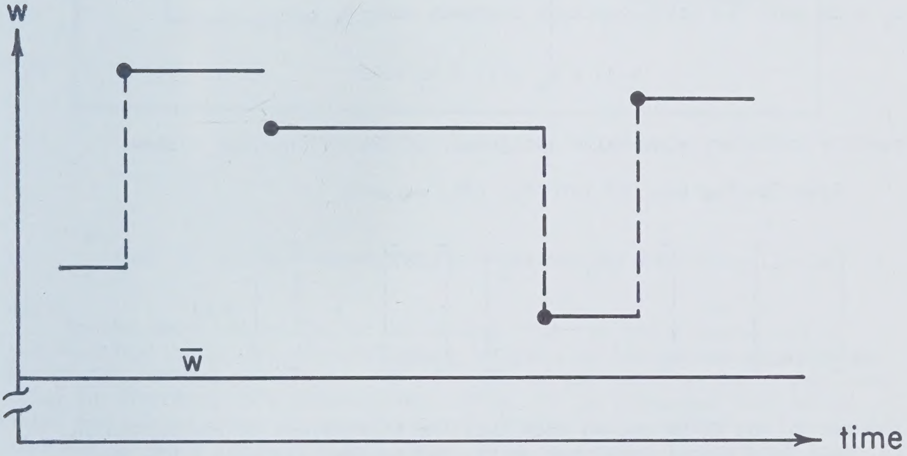


Fig. 2. Model for w(t).

the model for the accumulated error which is specified by

$$z(t+1) = z(t) + e(t) \tag{7}$$

The augmented system is then given by

$$\begin{bmatrix} e(t+1) \\ \\ z(t+1) \end{bmatrix} = \begin{bmatrix} I & 0 \\ \\ I & I \end{bmatrix} \begin{bmatrix} e(t) \\ \\ z(t) \end{bmatrix} + \begin{bmatrix} B \\ \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} G \\ \\ 0 \end{bmatrix} w(t) \tag{8}$$

and is 2n-dimensional where I is the n x n identity matrix. The controllability matrix for this system is

$$\begin{bmatrix} B & B & B & . & . & . \\ \\ 0 & B & 2B & . & . & . \end{bmatrix}$$

which is obviously of rank 2n if and only if B is of rank n. Hence, controllability of the original system implies controllability of the augmented one which, in turn, implies the existence of p x n matrices K_1 and K_2 such that the state-variable feedback control law

$$u(t) = K_1\, e(t) + K_2\, z(t) \tag{9}$$

permits arbitrary eigenvalue assignment of the closed-loop system.

Substituting Eq. (9) into Eq. (8), we get

$$\begin{bmatrix} e(t+1) \\ \\ z(t+1) \end{bmatrix} = \begin{bmatrix} I + BK_1 & BK_2 \\ \\ I & I \end{bmatrix} \begin{bmatrix} e(t) \\ \\ z(t) \end{bmatrix} + \begin{bmatrix} G \\ \\ 0 \end{bmatrix} w(t) \tag{10}$$

Now let K_1 and K_2 be chosen such that the eigenvalues of the system in Eq. (10) have moduli less than unity, and assume that $w(t) = w^*$, an unknown constant, for all t. Since the system is asymptotically stable in the sense of Lyapunov and bounded-input/bounded-output stable, it will attain an equilibrium (e^*, z^*) as $t \to \infty$. From the second line in Eq. (10), this equilibrium is characterized by $z^* = z^* + e^*$. Hence, $e^* = 0$ and we have zero steady-state, queueing error. We can achieve this condition in finite time, rather than asymptotically, by selecting K_1 and K_2 such that the 2n closed-loop eigenvalues of the system in Eq. (10) are all zero. In this case, we get "deadbeat response" [9, Ch. 6] which means that $e(t) = 0$ for some $t \leq 2n$ and all t thereafter. It is now clear why we assumed above that the jumps in w(t) are separated by more than 2nT units of time "on the average".

As an alternative to deadbeat response, one can choose K_1 and K_2 to place the system eigenvalues at specified points inside the unit circle in the complex plane for purposes of shaping the dynamic response, e.g., rise time and per cent overshoot, in driving toward zero queueing error. We shall not, however, pursue this here.

The routing algorithm that we have developed is defined by Eqs. (4), (7) and (9). Its implementation can be visualized as shown in the block diagram of Fig. 3. We observe from the figure that the algorithm is centralized (routing computations takes place at a single processor) and that global information on the network state is required by the processor. We discuss these points further in Section 4.

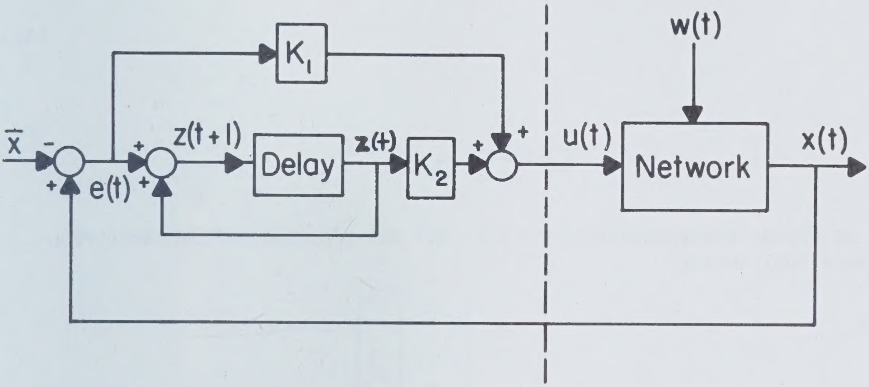


Fig. 3. Block diagram for feedback routing algorithm.

In the above algorithm, it is assumed that the state vector x(t), equivalently the vector of nodal queue lengths, is instantaneously available to the central processor. In a network of any size, we can expect some delay in obtaining this information. Hence, it is important that we examine the consequences of such "delayed information patterns" on the above approach to the routing problem. This we do in connection with the augmented model in Eq. (8) where we assume that z(t) which is internal to the processor is available instantaneously, but that e(t) is available only after L > 0 units of delay. We thus define the 2n-dimensional measurement vector

$$y(t) = \begin{bmatrix} z(t) \\ \\ e(t-L) \end{bmatrix} \tag{11}$$

The issue is now one of the observability of the system specified by Eqs. (8) and (11). We resolve this issue via a state augmentation approach which, while tractable, leads to a system of very high order. An alternative approach which results in a system of the same order as that in Eq. (8), but involves time-variable feedback of y(t) as defined in Eq. (11) is presented in [10].

To begin, we define

$$\xi_i(t) = e(t-i) \tag{12}$$

for i = 1, 2, ..., L. From this we get the set of additional state equations

$$\begin{aligned} \xi_1(t+1) &= e(t) \\ \xi_2(t+1) &= \xi_1(t) \\ &\vdots \\ \xi_L(t+1) &= \xi_{L-1}(t) \end{aligned} \tag{13}$$

If we append these equations to Eqs. (5) and (7), and define the (L+2)n-dimensional vector

$$\xi(t) = \begin{bmatrix} e(t) \\ z(t) \\ \xi_1(t) \\ \xi_2(t) \\ \cdot \\ \cdot \\ \cdot \\ \xi_L(t) \end{bmatrix}$$

we have the augmented system

$$\xi(t+1) = \bar{A}\xi(t) + \bar{B}u(t) + \bar{G}w(t) \tag{14}$$

where

$$\bar{A} = \begin{bmatrix} I & 0 & 0 & 0 & & & & 0 & 0 \\ I & I & 0 & 0 & & & & 0 & 0 \\ I & 0 & 0 & 0 & \cdot & \cdot & \cdot & 0 & 0 \\ 0 & 0 & I & 0 & & & & 0 & 0 \\ 0 & 0 & 0 & I & & & & 0 & 0 \\ & & \cdot & & & & & & \\ & & \cdot & & & & & & \\ & & \cdot & & & & & & \\ 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot & I & 0 \end{bmatrix}$$

$$\bar{B} = [B' \quad 0 \quad 0 \quad 0 \quad \cdot \quad \cdot \quad \cdot \quad 0]'$$

and

$$\bar{G} = [G' \quad 0 \quad 0 \quad 0 \quad \cdot \quad \cdot \quad \cdot \quad 0]'$$

In terms of the variables in Eqs. (12) and (13), Eq. (11) becomes

$$y(t) = \begin{bmatrix} z(t) \\ \\ \xi_L(t) \end{bmatrix} = \bar{C}\xi(t) \tag{15}$$

where $\bar{C}$ is 2n x (L+2)n and given by

$$\bar{C} = \begin{bmatrix} 0 & I & 0 & 0 & \cdot & \cdot & \cdot & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot & 0 & I \end{bmatrix}$$

The system of interest is defined by the triple $(\bar{A}, \bar{B}, \bar{C})$. Via direct calculation, its controllability matrix is

$$\begin{bmatrix} B & B & B & B & & & & B & & & \\ 0 & B & 2B & 3B & & & & [(L+2)n-1]B & & & \\ 0 & B & B & B & \cdot & \cdot & \cdot & B & \cdot & \cdot & \cdot \\ 0 & 0 & B & B & & & & B & & & \\ 0 & 0 & 0 & B & & & & B & & & \\ & \cdot & & & & & & & & & \\ & \cdot & & & & & & & \cdot & \cdot & \cdot \\ & \cdot & & & & & & & & & \\ 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot & B & \cdot & \cdot & \cdot \end{bmatrix}$$

The full controllability matrix is (L+2)n x (L+2)np, but it is sufficient for purposes of analysis to truncate its evaluation as indicated. It is easy to see that this matrix is of rank (L+2)n and therefore that the system in Eq. (14) is controllable if and only if rank B = n as assumed earlier. However, since only $z(t)$ and $\xi_L(t)$ can be measured, an observer must be used to reconstruct $\xi(t)$ for purposes of state-variable feedback in order to get zero steady-state queuing error as for the system in Eq. (10).

The observability matrix of the pair $(\bar{A}, \bar{C})$ is

$$\left[\begin{array}{cccccccccc}
0 & I & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & & 0 & 0 & 0 & 0 & I \\
\hline
I & I & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & & 0 & 0 & I & 0 & 0 \\
\hline
2I & I & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & & 0 & 0 & I & 0 & 0 \\
\hline
3I & I & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & & 0 & I & 0 & 0 & 0 \\
\hline
 & & \vdots & & \vdots & & & \vdots & &
\end{array}\right]$$

The full observability matrix is $2(L+2)n^2 \times (L+2)n$, but as above in the case of controllability, its evaluation can be truncated as shown. This matrix is clearly of rank (L+2)n so that the system defined by Eqs. (14) and (15) is observable. Since $z(t)$ and $\xi_L(t)$ are measured directly, the least-order observer for the reconstruction of $\xi(t)$ has order nL. This analysis shows that delayed information patterns pose no conceptual difficulty. However, dimensionality in state reconstruction is of some concern if L is of any size at all.

4. CONCLUSION

We have presented certain aspects of the state-space approach for the modeling of information flow in data-communication networks, and shown the application of multivariable control theory to the routing problem in such networks. Of particular interest are the results on (1) controllability and observability, and the preservation of these properties in the presence of delayed information patterns, and (2) the derivation of stable, feedback routing policies which produce zero steady-state queueing error, and permit arbitrary shaping of the network's dynamic response via closed-loop eigenvalue assignment.

The present work also serves to define some important and interesting problems that deserve further study. The first of these is that of controllability. The analysis here is classical and allows u(t) to take on negative values. Strictly speaking, this is not permissible since $u_{ij}^{k}(t)$ is a directed flow. Hence, the analysis needs to be reconsidered subject to $u(t) \geq 0$. It should likewise include the effects of channel capacity constraints. A second problem of interest has to do with the high dimensionality of the feedback routing algorithms and the need for full state-variable feedback with either instantaneous or delayed information patterns. This certainly warrants studying potential applications of decentralized control and delayed local (e.g., adjacent node) information patterns in developing routing strategies. Finally, a third problem is that of specifying other performance criteria such as average message delay and throughput, and developing feedback routing strategies that minimize the former and maximize the latter. These two criteria, while readily amenable to the queueing theory approach to network optimization [6,7], do not appear immediately tractable here in the absence of additional assumptions about the model.

REFERENCES

1. A. Segall and F. H. Moss, "Application of optimal control theory to dynamic routing for computer networks," Proc. 1976 Joint Autom. Control Conf., Purdue Univ., W. Lafayette, IN, July 1976, pp. 541-546.
2. F. H. Moss and A. Segall, "Progress in the application of optimal control theory to dynamic routing in data communication networks," Proc. 1976 IEEE Conf. on Decision and Control, Clearwater Beach, FL, December 1976, pp. 954-956.
3. A. Segall, "The modeling of adaptive routing in data-communication networks," IEEE Trans. Commun., vol. COM-25, pp. 85-95, January 1977.
4. J. S. Meditch, "Multivariable adaptive routing in data-communications networks," Proc. 1977 Natl. Telecomm. Conf., Los Angeles, CA, December 1977, to appear.
5. __________, "Stochastic regulation of queues in data networks," Proc. 1977 IEEE Conf. on Decision and Control, New Orleans, LA, December 1977, to appear (invited paper).
6. L. Kleinrock, Communication Nets: Stochastic Message Flow and Delay. New York: McGraw-Hill, 1964.
7. __________, Queuing Systems, Volume 2: Computer Applications. New York: Wiley-Interscience, 1976.
8. C. A. Wolfe and J. S. Meditch, "Theory of system type for linear multivariable servomechanisms," IEEE Trans. Automat. Contr., vol. AC-22, pp. 36-46, February 1977.
9. H. Kwakernaak and R. Sivan, Linear Optimal Control Systems. New York: Wiley-Interscience, 1972.
10. J. S. Meditch and A. T. Fam, "On the role of slow and multiplexing controllability and observability in computer-communication networks," to appear.

ROUTING TO MULTIPLE SINKS IN A COMMUNICATION NETWORK

CHARLES A. WOLFE
HAROLD J. PAYNE
Orincon Corporation
La Jolla, California 92037

CHARLES L. MOREFIELD
BRUCE W. GARDNER

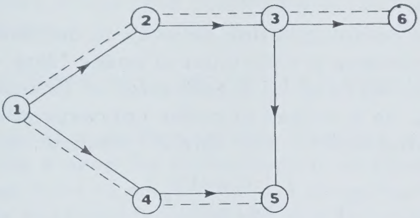
ABSTRACT

This paper discusses routing in a communication network where the information from each of several sources is sent to multiple sinks. The criteria to be used in the routing problem is a minimal use of available communication channel capacity. Assuming that data can be replicated at a communication node for transmission on any of the outgoing links, the notion of a connecting tree becomes central to this problem. A suboptimal method of solution, which involves generation of a set of candidate connecting trees corresponding to each source, is detailed here and illustrated by several examples.

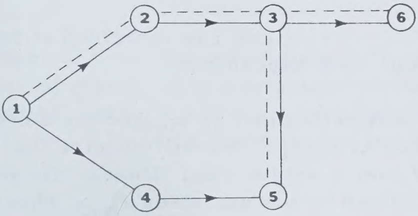
1. INTRODUCTION

In communications networks which provide for the transmission of data from (remote) data sources to any of several users, the simultaneous provision of data from a data source to multipe sinks (or users) creates a new kind of routing problem. To see the nature of this problem, consider the graph and routes illustrated in figures 1a and 1b. In figure 1a, the data flows from the source node ① to the two sink nodes, ⑤ and ⑥ , along two distinct paths, each of which is a corresponding shortest path. This routing requires five units of communication capacity, as five links in all are required. Now suppose that data can be replicated enroute. Replication has been made possible in communications networks by the development of substantial computer processing power in nodes. Then, as illustrated in figure 1b, the data transmission requirements can be achieved by the use of a tree with branches at nodes intermediate between the source and the several sinks. Here, we see that this routing, with data replication at node ③ , requires four rather than five units of communication capacity.

Where one source is concerned, the routing problem can be phrased as the identification of a "minimal connecting tree", a notion similar to the standard notion of a minimal spanning tree.[1] A spanning tree is a subset of links such that for each pair of nodes there is a unique connecting path using only links in the tree. A connecting tree differs in that connections are of relevance only for certain pairs of nodes. In this instance, for example, we are only concerned with the connectivities between the source and each of the several sinks. Given costs associated with links, the cost (or length) of a tree is the sum of the costs corresponding to the links included in the tree. A minimal connecting (spanning) tree is a connecting (spanning) tree of minimal cost (or length). There exist efficient algorithms for generating minimal spanning trees [2]; there do not appear to be corresponding algorithms available for generating minimal connecting trees. We have based our approach on a use of a minimal spanning tree algorithm.



(a) Shortest paths from ① to ⑤ and ⑥



(b) Path with message replicating node from ① to ⑤ and ⑥

Figure 1. Shortest paths and shortest trees.

In identifying a connecting tree for a source, it is of course necessary that the capacity of each link included be sufficient to handle the data stream. This may preclude the use of the minimal connecting tree. Hence, the complete problem statement is seen to involve attention to link capacity constraints.

Where several sources are concerned, a connecting tree must be determined for each source. The issue of link capacity constraints creates an interdependence in the selections for each source, and considerably complicates the problem. In the next section, we provide a detailed formulation of this capacity-constrained, multiple-source, connecting tree selection problem. Ideally, this formulation entails the enumeration of all possible connecting trees for each source. For many network problems, this leads to a prohibitively large collection of trees. As a result, it is desirable to pursue alternative approaches. We have developed a suboptimal procedure, detailed in section 3, in which a subset of the possible connecting trees for each source is generated, and the optimal simultaneous selection from these subsets is made.

The application of this suboptimal approach is illustrated in examples presented in section 4. This approach represents only a first attempt to resolve the full problem stated in section 2. Some notions as to potential avenues to improved approaches are presented in concluding comments, section 5.

2. FORMULATION OF THE PROBLEM

The structure of a communication network is defined by a graph, $G = \{N, L\}$, where N denotes a collection of nodes (data sources, intermediate switches and sinks) and L, a collection of (transmission) links. Let $N_k \subset N$, $k=1, \ldots, K$, be a subset of nodes corresponding to the kth source and its associated sinks. For this kth source, denote by

$$T_k = \{t_{kp}\}_{p=1}^{p=|T_k|}$$

the set of connecting trees. To define these trees, let a_{ikp} be defined as follows

$$a_{ikp} = \begin{cases} 1 & \text{if link } i \text{ is contained in tree } t_{kp} \\ 0 & \text{otherwise} \end{cases}$$

The capacity of link i will be denoted by c_i, and the amount of data being transmitted from source k, by q_k. We will require that all data from one source be transmitted over a single tree. Hence, the selection of connecting trees can be defined by the variables x_{km}, where

$$x_{km} = \begin{cases} 1 & \text{if tree } t_{km} \text{ is selected for source } k \\ 0 & \text{otherwise} \end{cases}$$

The restriction of a single tree for each source is quantified by the equation

$$\sum_{p=1}^{|T_k|} x_{kp} = 1, \quad k = 1, \ldots, K \tag{2.1}$$

The link capacity constraint is expressed by

$$\ell_i \equiv \sum_{k=1}^{K} \sum_{p=1}^{|T_k|} a_{ikp}\, x_{kp}\, q_k \le c_i, \quad i = 1, \ldots, |L| \tag{2.2}$$

In selecting the connecting trees, the objective will be to minimize the total communication load, measured by the sum of link loads over all links, i.e.,

$$S = \sum_{i=1}^{|L|} \ell_i \tag{2.3}$$

The problem as stated here is a 0-1 integer linear program, and methods exist for its solution.[3] A significant problem would arise in many applications in enumerating all the connecting trees as this formulation requires. And, even were one to perform this enumeration, the number of such trees would lead to enormous, and perhaps prohibitive, computational expense.

3. A SUBOPTIMAL METHOD OF SOLUTION

The computational problems arising from the enumeration of connecting trees create a need for consideration of alternative approaches. The approach taken here is one in which a compromise in optimality is made to achieve reasonable computational efficienty. Specifically, we retain the problem as formulated in the previous section, except that the set of connecting trees corresponding to each source is limited to a subset of those which are possible. An optimal selection from these subsets is made.

First, we describe the method of solution of the problem with specified collections of trees. It is convenient to reformulate the problem. For this purpose, let $y_k = p$ if tree p for source k is selected. This corresponds to $x_{kp} = 1$, and eliminates the need for equation (2. 1). Next define

$$b_{ik} = a_{ikp} \quad \text{if} \quad y_k = p \tag{3. 1}$$

The link capacity constraints are now expressed as

$$\ell_i = \sum_{k=1}^{K} b_{ik}\, q_k \le c_i, \; i = 1, \ldots, |L| \tag{3. 2}$$

The objective remains as before, i. e., to minimize

$$S = \sum_{i=1}^{|L|} \ell_i \tag{3. 3}$$

In this form, the problem is amenable to solution by a standard branch-and-bound technique.[2] In this method, a partial solution consisting of a specification of the first m components of $y = (y_1, \ldots, y_K)$ can be examined for feasibility with respect to the link capacity constraints, i. e., one requires

$$\sum_{k=1}^{m} b_{ik}\, q_k \le c_i, \; i = 1, \ldots, |L|, \; m \le k \tag{3. 4}$$

and a bound on S can be generated. For this latter purpose, we identify, for each k, the least costly tree, having cost

$$\bar{\lambda}_k \equiv \max_{p} \sum_{i=1}^{|L|} a_{ikp}.$$

Then

$$S = \sum_{i=1}^{|L|} \sum_{k=1}^{K} b_{ik}\, q_k \geq \sum_{k=1}^{m} b_{ik}\, q_k + \sum_{k=m+1}^{K} \bar{\lambda}_k \equiv \bar{S}$$

A partial solution need not be further considered if $\bar{S}$ exceeds the value associated with a previously determined candidate solution.

Now we consider the matter of generating subsets of the possible connecting trees for each source. The method described here is but one of a number of such methods that could be identified. No claim for optimality in this part of the method is made.

The method is based upon a use of an available, efficient algorithm for computing minimal spanning trees. We note that a connecting tree can be determined from a spanning tree by pruning unnecessary nodes and links. To see how this is done, consider figure 2. The dashed line indicates a link and node which are eliminated from the spanning tree to yield the desired connecting tree.

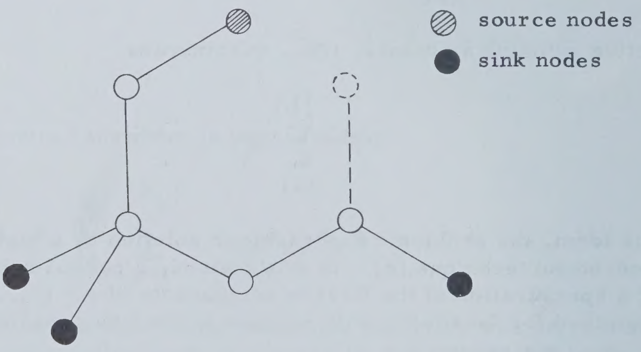


Figure 2. Pruning a spanning tree to obtain a connecting tree.

Let M(G, D), where G denotes the graph and D the set of link costs, denote the minimal spanning tree problem, with solution tree t(G, D). The pruning process, given N_k, yields the connecting tree, $\bar{t}\,[t(G, D), N_k]$. The algorithm for generating connecting trees is based on sequentially modifying the link costs for links included in connecting trees generated to induce the generation of different minimal spanning trees.

Algorithm (Generation of Connecting Trees)

```
Step 1:  Set m = 0
         Specify $D = \left\{d_i\right\}_{i=1}^{i=|L|}$   (link costs)
         Specify $\delta$ (link cost increment)
Step 2:  Solve M(G, D) to obtain the minimal spanning tree, t
Step 3:  Prune t to accomodate nodes in $N_k$ to obtain the
         connecting tree $\bar{t}$
         If $\bar{t} \neq t_{kp}$, p = 1, ..., m,
                 Set m = m + 1
                 Set $t_{km} = \bar{t}$
                 Set $a_{ikm}$ = 1 if link i is contained in $t_{km}$
                               and 0 otherwise
                 If m = P, stop
Step 4:  Set $d_i = d_i + \delta a_{ikm}$
         Go to Step 2.
```

4. EXAMPLES

For an example, we consider the graph of figure 3, with link costs as indicated. With these link costs, nodes ①, ⑥ and ⑨ will not serve as intermediate nodes in any tree; they can only be terminal nodes. In this instance, this is a device to reflect the fact that links exist to these nodes but not from these nodes.

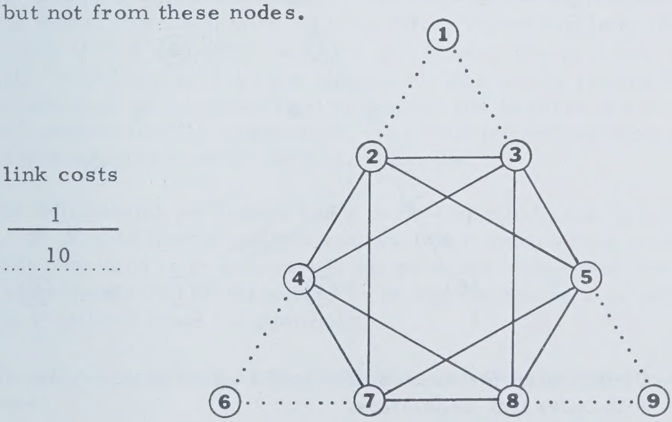


Figure 3. Graphs with link costs for example.

We consider transmission from three sources, as follows:

From node ⑧ to nodes ① and ⑥
From node ④ to nodes ① and ⑨
From node ③ to nodes ⑥ and ⑨

Five connecting trees were generated for each source. With (1,2) devoting the link from node ① to node ② , these trees are as follows:

For source at node ⑧ :

Tree 1: (8,4), (4,6), (4,2), (2,1)
Tree 2: (8,3), (3,1), (8,7), (7,6)
Tree 3: (8,7), (7,6), (7,2), (2,1)
Tree 4: (8,3), (3,1), (8,4), (4,6)
Tree 5: (8,4), (4,6), (4,3), (3,1)

For source at node ④ :

Tree 6: (4,8), (8,9), (4,2), (2,1)
Tree 7: (4,3), (3,1), (3,5), (5,9)
Tree 8: (4,3), (3,1), (4,2), (2,5), (5,9)
Tree 9: (4,3), (3,1), (4,7), (7,5), (5,9)
Tree 10: (4,8), (8,5), (5,9), (4,3), (3,1)

For source at node ③ :

Tree 11: (3,5), (5,9), (3,4), (4,6)
Tree 12: (3,8), (8,9), (8,7), (7,6)
Tree 13: (3,4), (4,6), (3,8), (8,9)
Tree 14: (3,5), (5,9), (5,7), (7,6)
Tree 15: (3,4), (4,6), (4,8), (8,9)

With all link capacities equal to unity, and all source data rates equal to one-half, the solution found is to use trees 1, 6, 11 (see figure 4).

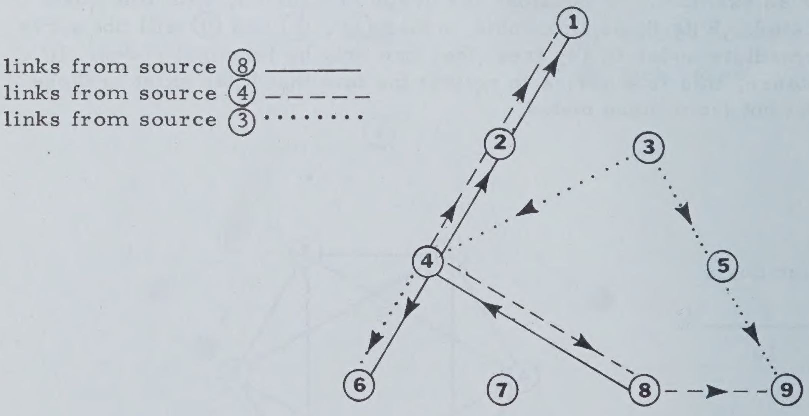


Figure 4. Circuit plan for source data rates equal to one-half, with unity link capacities.

With the source data rates increased to unity, the solution found is to use trees 1, 7, 12 (see figure 5).

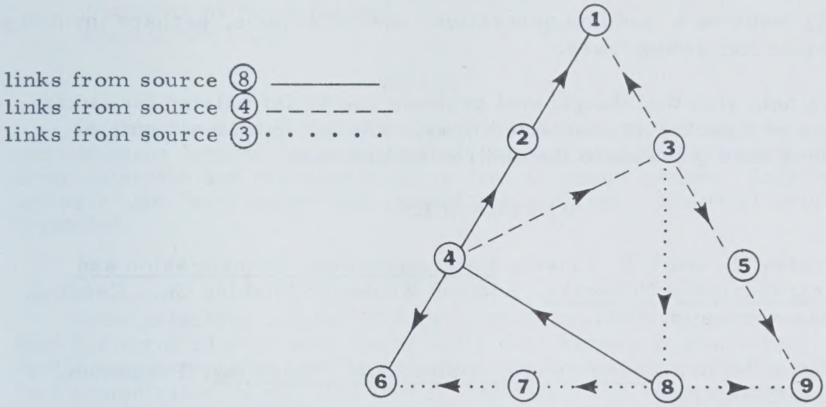

links from source ⑧ ————
links from source ④ — — — — —
links from source ③ · · · · · ·
1
2
3
4
5
6
7
8
9

Figure 5. Circuit plan for source data rates equal to unity, with unity link capacities.

We note at this point that though the problem is symmetric, the suboptimal algorithm did not select connecting trees which reflected this symmetry (e.g., for source at node ④ , some trees have length 5, all others are length 4).

5. CONCLUDING COMMENTS

This paper has described a routing algorithm for networks that have multiple sinks for each data source with data being distributed over connecting trees. This type of routing is made possible by the use of "smart" nodes which are capable of sending the same data on multiple outgoing links. The objective of the routing algorithm is to minimize the amount of link capacity used. In this paper, the problem was formulated as an 0-1 integer, linear program. Due to the generally very large number of connecting trees involved, the problem as formulated may be computationally intractable, so a suboptimal solution procedure was formulated and used to solve an example.

The suboptimal procedure used proved adequate for our immediate needs. We would like to point out some other approaches to the problem. The first approach is to enumerate all possible connecting trees. This would be possible for small networks or for moderate size networks if there is a critical need for optimality.

The second approach is to develop an iterative scheme consisting of two steps:

(1) solution of the optimal tree selection problem for prescribed sets of trees for each source (e.g., as done in this paper), and

(2) generation of new trees based upon the results of step (1).

Step (2) would be a "column generation" sort of scheme, perhaps involving k-shortest connecting trees.

We note also that the general problem can be formulated for either the case of directed or undirected links. The suboptimal algorithm presented here pertains to the undirected link case.

REFERENCES

1. Frank, H. and I. T. Frisch, Communication, Transmission and Transportation Networks, Addison Wesley Publishing Co., Reading, Massachusetts, 1971.

2. Wells, M. B., Elements of Combinatorial Computing, Pergamon, New York, 1971.

3. Garfinkel, R.S. and G. L. Nemhauser, Integer Programming, Wiley, New York, 1972.

ON THE STATISTICAL ANALYSIS OF CONCENTRATORS WITH ARQ RETRANSMISSION SCHEMES

DON TOWSLEY AND JACK KEIL WOLF
Electrical and Computer Engineering Department
University of Massachusetts
Amherst, MA 01003

ABSTRACT

In this paper we derive formulae for the average waiting times and average queue lengths for both slotted and unslotted concentrators when error detection and retransmission is used to combat errors. Both "stop and wait" and "continuous" ARQ schemes are analyzed. Numerical results are presented.

INTRODUCTION

Error detection coupled with retransmission (ARQ) is a common method used for error control when transmitting data between concentrators. In such a scheme messages must be stored at the sending concentrator until that concentrator is notified that retransmission is not required. This storage of messages after initial transmission affects the queue lengths and waiting times for the concentrator.

In this paper we analyze two ARQ schemes: "stop-and-wait ARQ" and "continuous ARQ". In the stop-and-wait strategy, the sending concentrator transmits a message block and then waits for a return message indicating whether or not the block was received correctly. If a positive acknowledgement (an ACK) is received, that message block is purged from the sending concentrator's buffer and the concentrator is ready to transmit the next message block. If a negative acknowledgement (a NACK) is received, or if errors are detected in an ACK/NACK message, that message block is retransmitted and the process repeats. It should be noted that the sending concentrator is idle during the time it is waiting for the ACK/NACK response. In the continuous ARQ strategy, this idle period is eliminated since the sending concentrator continuously transmits message blocks without waiting for an acknowledgement. Again ACKs and NACKS are sent over a return channel but when a NACK arrives for a particular message block, that message block and all subsequent message blocks are retransmitted. The receiver does not send a positive acknowledgement for any message block until it has sent positive acknowledgements for all previous message blocks.

In this paper we develop formulae for the average queue lengths and the average waiting times for slotted and unslotted concentrators utilizing either of these ARQ strategies. In particular we show how these schemes can be reduced to special cases of models for non ARQ schemes which are easily analyzed by standard techniques. Although there have been a number of papers concerned with this subject, [1,2,3] the results presented here, to the knowledge of the authors, have not been explicitly presented before.

SLOTTED CONCENTRATOR

In a slotted concentrator without ARQ, [4] the cencentrator transmits one message block every Δ seconds while the queue is non-empty. Each message block is assumed to be of duration Δ seconds. If a message block arrives to an empty queue, it is transmitted in the next slot of duration Δ. We assume that the number of arrivals in any slot is statistically independent of the number of arrivals in any other slot with mean μ_D and variance

σ_D^2. Let L_i be the number of message blocks stored in the queue just prior to the end of the i^{th} slot. Let D_i be the number of message blocks which arrived during that i^{th} slot. Then we have

$$L_i = L_{i-1} - U(L_{i-1} - 1) + D_i \ . \tag{1}$$

Here $U(x)$ is the unit step defined as: $U(x) = 0$, $x < 0$; $U(x) = 1$, $x \geq 0$. When the concentrator is in equilibrium, it has been shown [5] that the average number of message blocks in the queue is

$$E[L] = \frac{\mu_D}{2} + \frac{\sigma_D^2}{2(1-\mu_D)} \quad \text{message blocks,} \tag{2}$$

while the average waiting time is

$$E[W] = \frac{1}{2} + \frac{\sigma_D^2}{2\ \mu_D(1-\mu_D)} \quad \text{slots.} \tag{3}$$

We now consider a slotted concentrator with stop and wait ARQ. Assume that M_j is a random variable, denoting the number of slots required to successfully transmit the j^{th} message block. Let successive M_j be statistically independent and identically distributed with means μ_M and variance σ_M^2. If p is the probability of error in the single transmission of a message block (whether it is the initial transmission or a retransmission), if p' is the probability of error in an ACK/NACK message, if all errors are independent, and if s is the number of slots in the idle period, then μ_M and σ_m^2 are given as

$$\mu_M = (1+s)\ [\frac{1}{1-p} + \frac{p'}{1-p'}], \tag{4}$$

$$\sigma_M^2 = (1+s)^2\ [\frac{p}{(1-p)^2} + \frac{p'}{(1-p')^2}] \ . \tag{5}$$

We now create a fictitious quantity called a slacket. When the j^{th} message block enters the queue it is assigned M_j such slackets, this number equal to the number of slots required to successfully transmit that message block. Let $\mathcal{L}_i$ denote the number of slackets stored in the queue just prior to the end of the i^{th} slot and let $\mathcal{D}_i$ be the number of slackets which arrived during that i^{th} slot. Then

$$\mathcal{L}_i = \mathcal{L}_{i-1} - U(\mathcal{L}_{i-1} - 1) + \mathcal{D}_i \ . \tag{6}$$

Equation (6) is of the same form as (1) except that it concerns slackets instead of message blocks. Thus when the concentrator is in equilibrium, the average number of slackets stored in the queue is

$$E[\mathcal{L}] = \frac{\mu_{\mathcal{D}}}{2} + \frac{\sigma_{\mathcal{D}}^2}{2(1-\mu_{\mathcal{D}})} \ . \tag{7}$$

Equation (7) gives the average amount of unfinished work remaining in the queue at a time prior to a slot boundary.

It remains to relate the parameters in (7) to the known parameters μ_D, σ_D^2, μ_M, σ_M^2 and the desired parameter μ_L. Since $\mathcal{D}$ is the sum of D statistically independent, identically distributed random variables M, then

$$\mu_{\mathscr{D}} = \mu_D \, \sigma_M \; ; \tag{8}$$

$$\sigma_{\mathscr{D}}^2 = \mu_D \, \sigma_M^2 + \mu_M^2 \, \sigma_D^2 . \tag{9}$$

Substituting in (7) we obtain

$$E[\mathscr{L}] = \frac{\mu_D \, \mu_M}{2} + \frac{\mu_D \, \sigma_M^2 + \mu_M^2 \, \sigma_D^2}{2(1 - \mu_D \, \mu_M)} \; . \tag{10}$$

Obtaining E[L] from $E[\mathscr{L}]$ is more difficult since if the concentrator is not empty, the number of slackets in the message block being transmitted has a distribution which differs from the distribution of the random variable M. In general,

$$\mathscr{L} = M_1 + M_2 + \cdots + M_{L-1} + \mathscr{M} \, U(L) \; , \tag{11}$$

where $\mathscr{M}$ is the random variable denoting the remaining number of slackets for the message block currently being transmitted. Here $\mathscr{M}$ is the random variable for the residual service time at a random point in time and has mean $\mu_{\mathscr{M}}$ given by

$$\mu_{\mathscr{M}} = \frac{\mu_M + 1}{2} + \frac{\sigma_M^2}{2\mu_M} \; . \tag{12}$$

Taking the expectation of both sides of (11), noting that L and M_i are independent, we obtain

$$E[L] = \frac{\mu_D \, \mu_M}{2} + \frac{\mu_D^2 \, \sigma_M^2 + \mu_M \, \sigma_D^2}{2(1 - \mu_D \, \mu_M)} \; . \tag{13}$$

Utilizing Little's formula [6], we obtain the average waiting time, E[W]

$$E[W] = \frac{E[L]}{\mu_D} . \tag{14}$$

We next consider the slotted concentrator with continuous ARQ. We continue the fiction of message blocks being converted to slackets prior to their entrance in the concentrator. We begin by analyzing a modified continuous ARQ system slightly different from the actual system. In this modified system, a message block is removed from the queue immediately after its last transmission. (In the actual system, the message block is removed s slots later where s is the round trip delay measured in slots.) The effect of this assumption is that the service intervals for the message blocks do not overlap so that a given slot can be assigned to no more than one message block. Let M_j be the number of slots required to transmit the j^{th} message block for the modified system. Then if p is the probability of a block error in the forward direction, if p' is the probability of error in an ACK/NACK message, if all errors are statistically independent, then the mean and variance of M are

$$\mu_M = \frac{(1 + s)p}{1 - p} + \frac{(1 + sp')}{1 - p'} \; , \tag{15}$$

and

$$\sigma_M^2 = (1 + s)^2 \left[\frac{p}{(1-p)^2} + \frac{p'}{(1-p')^2} \right] . \tag{16}$$

It is easily argued that (13) gives the average occupancy of the modified system where μ_M and σ_M^2 and given as in (15) and (16). We now solve for the average occupancy of the correct problem. Let L be the occupancy of the true system and let K be the occupancy of the modified system.

Since, on the average, s μ_D message blocks arrive in the s extra slots that the true system must hold the message block, we have

$$E[L] = E[K] + s\,\mu_D \,. \tag{17}$$

Thus the average queue length for the true system is

$$E[L] = \frac{\mu_D}{2}\,[\mu_M + 2s] + \frac{\mu_D^2\,\sigma_M^2 + \mu_M\,\sigma_D^2}{2(1 - \mu_M\,\mu_D)} \,, \tag{18}$$

where μ_M and σ_M^2 are given by (15) and (16). Using Little's formula we obtain the average delay to the receipt of the ACK for the message block

$$E[W] = \frac{(\mu_M + 2s)}{2} + \frac{\mu_D^2\,\sigma_M^2 + \mu_M\,\sigma_D^2}{2\,\mu_D(1 - \mu_M\,\mu_D)} \,. \tag{19}$$

Numerical results for E[L] for the slotted concentrator are given in Figures 1 and 2 for the stop and wait and continuous systems respectively. For both systems, it is assumed that $\mu_D = \sigma_D^2 = \lambda$ (which applies to Poisson arrivals) and that $p = p'$. In Figure 1, the curves are parametized by the single parameter $(1 + s)\,\lambda = \lambda'$. In Figure 2, two parameters s and λ are required.

UNSLOTTED CONCENTRATOR

In the unslotted concentrator there are no slots to which message transmission is synchronized. Messages arriving to an empty queue are immediately transmitted. We will assume that messages may have an arbitrary length X with probability density functions $f_X(x)$, mean μ_X and variance σ_X^2. Message arrivals are assumed to be Poisson where λ is the average number of messages arriving per unit time.

For an unslotted concentrator, let N_j be the number of messages in the concentrator immediately after the completion of the transmission of the j^{th} message. Let A_j be the number of messages that arrived during the service time of the j^{th} message (excluding the j^{th} message). The following relation describes the behavior of the system,

$$N_j = N_{j-1} - U(N_{j-1} - 1) + A_j . \tag{20}$$

The equilibrium solution to this equation for E[N] is

$$E[N] = \frac{E[A]}{2} + \frac{\sigma_A^2}{2(1 - E[A])} \,. \tag{21}$$

Here E[A] and σ_A^2, the mean and variance of A, are given as

$$E[A] = \lambda\,\mu_X , \tag{22}$$

$$\sigma_A^2 = \lambda\,\mu_X + \lambda^2\,\sigma_X^2 . \tag{23}$$

It has been shown that the Pollaczek-Khinchin formula is also valid when viewing the system at a random point in time.

Let us now consider the unslotted concentrator with stop and wait ARQ. Let Y be a random variable denoting the delay a message incurs waiting for an ACK/NACK message with mean μ_Y and variance σ_Y^2. Let $N_j(X)$ be a random variable which denotes the number of transmissions that the j^{th} message requires until successfully transmitted and acknowledged with mean $\mu_{N(X)}$ and variance $\sigma_N^2(X)$. Usually $N_j(X)$ will depend upon the length of a message. Finally, let χ_j be a random variable denoting the time required to successfully transmit message j from its initial transmission to receipt of its final ACK. Under the assumption of errors as independent events, the unslotted concentrator with stop and wait ARQ can be viewed as a M/G/1 system with average queue length given by

$$E[N] = \frac{\lambda\mu_\chi}{2} + \frac{\lambda\mu_\chi + \lambda^2\sigma_\chi^2}{2(1 - \lambda\mu_\chi)} \tag{24}$$

and average waiting time

$$E[W] = E[N]/\lambda . \tag{25}$$

It is easy to show that the mean μ_χ and variance σ_χ^2 are given by

$$\mu_\chi = \int_0^\infty \mu_{N(X)}(x + \mu_Y)\, f_X(x)dx, \tag{26}$$

and

$$\sigma_\chi^2 = \int_0^\infty [(x + \mu_Y)^2(\sigma_N^2(X) + \mu^2_{N(X)}) + \mu_{N(X)}\sigma_Y^2]f_X(x)dx - \mu_\chi^2 . \tag{27}$$

For the special case where $\sigma_X^2 = 0$ (i.e., constant message lengths), then the probability of error can be taken as a constant p. If p' is the probability of error in an ACK/NACK message, if all errors are independent, then μ_N and σ_N^2 are given by

$$\mu_N = \frac{1}{1 - p} + \frac{p'}{1 - p'} , \tag{28}$$

$$\sigma_N^2 = \frac{p}{(1 - p)^2} + \frac{p'}{(1 - p')^2} . \tag{29}$$

These expressions can be substituted into (26) and (27) to obtain

$$\mu_\chi = (\mu_X + \mu_Y)\left[\frac{1}{1 - p} + \frac{p'}{1 - p'}\right], \tag{30}$$

and

$$\sigma_\chi^2 = (\mu_X + \mu_Y)^2\left[\frac{p}{(1 - p)^2} + \frac{p'}{(1 - p')^2}\right] + \sigma_Y^2\left[\frac{1}{1-p} + \frac{p'}{1-p'}\right] . \tag{31}$$

For the unslotted concentrator with continuous ARQ, again it is convenient to first treat a modified system where the message leaves the queue immediately after its last transmission (and before receipt of the ACK). The average length of the queue for this modified system is also given by (24). To obtain the average queue length for the true system one must add to (24) the average number of messages that arrive in the interval between the end of the last transmission of a message and the receipt of the ACK/NACK message. Under the same assumptions that led to (30) and (31) we now have

$$\mu_\chi = \frac{(\mu_X + \mu_Y)p}{1 - p} + \frac{\mu_X + \mu_Y\, p'}{1 - p'} , \tag{32}$$

and

$$\sigma_\chi^2 = (\mu_X + \mu_Y)^2\left[\frac{p}{(1 - p)^2} + \frac{p'}{(1 - p')^2}\right] + \sigma_Y^2\left[\frac{p}{1-p} + \frac{p'}{1-p'}\right] . \tag{33}$$

These can be substituted into the following expressions to obtain the mean queue length,

$$E[N] = \frac{\lambda\mu_X + 2\mu_Y}{2} + \frac{\lambda\mu_X + \lambda^2\sigma_X^2}{2(1 - \lambda\mu_X)} \tag{34}$$

and average waiting time

$$E[W] = E[N]/\lambda, \tag{35}$$

for the continuous ARQ system.

Numerical results for the unslotted concentrator for E[L] are given in Figures 3-6. In all cases it is assumed that $p = p'$ and $\mu_X = 1$. Figures 3 and 4 present curves for the stop and wait system for the case $\sigma_Y^2 = 0$ and $\sigma_Y^2 = \mu_Y^2$ respectively. Similarly, Figures 5 and 6 present curves for the continuous system for the case $\sigma_Y^2 = 0$ and $\sigma_y^2 = \mu_Y^2$ respectively. In Fig.3 the curves are parameterized by the parameter $(\mu_X + \mu_Y)\lambda = \lambda'$. In Figures 4-6, two parameters λ and μ_Y are required.

THROUGHPUT

An important parameter of any data communications system is its throughput. For a slotted ARQ system, the throughput can be defined as the ratio of the average number of message packets successfully transmitted to the number of slots required to accomplish this transmission. For the two retransmission schemes analyzed here the throughput is

Stop and Wait ARQ $\quad \text{Throughput} = [1+s]^{-1} \left[\frac{1}{1-p} + \frac{p'}{1-p'}\right]^{-1}$

Continuous ARQ $\quad \text{Throughput} = \left[\frac{(1+s)p}{1-p} + \frac{(1+s\,p')}{1-p'}\right]^{-1}$

Numerical results for these equations are given in Figure 7 for the case where $p' = 0$.

REFERENCES

[1] Chu, W. W., "Optimal Message Block Size for Computer Communications with Error Detection and Retransmission Strategies," IEEE Trans. on Communications 22, 1516 (1974).

[2] Schweitzer, P. J. and S. S. Lam, "Buffer Overflow in a Store-and-Forward Network Node," IBM J. of Res. and Develop. 20, (1976).

[3] Gavish, B. and A. G. Konheim, "Computer Communication Via Satellites - A Queueing Model," IEEE Trans. on Communications 25, 140 (1977).

[4] Schwartz, M. Computer-Communication Network Design and Analysis, Prentice Hall, New Jersey, 1977.

[5] Chu, W. W. and A. G. Konheim, "On the Analysis and Modeling of a Class of Computer Communications Systems," IEEE Trans. on Communications 20, 645 (1972).

[6] Little, D. C., "A Proof of the Queueing Formula: L = λW," Operations Research, 383 (1961).

[7] Kleinrock, L., Queueing Systems, Vol. 1, Wiley-Interscience, New York, 1975.

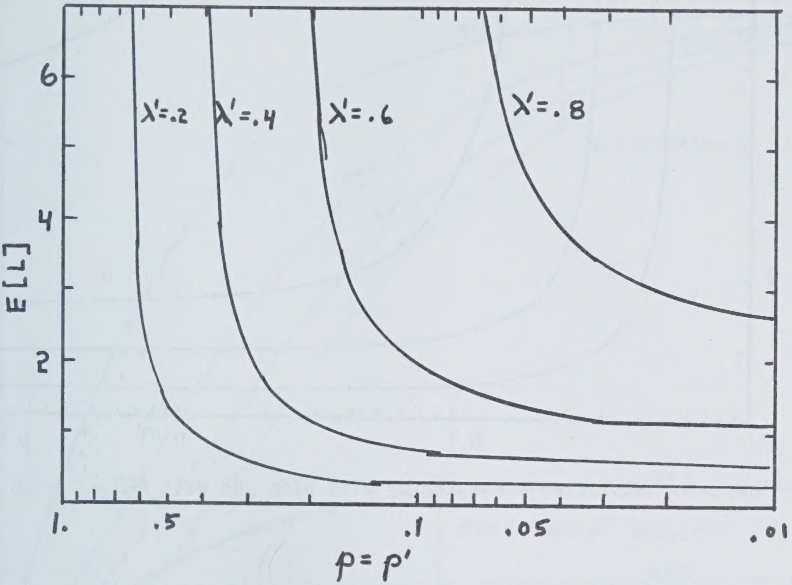


Figure 1. Average Queue Length versus p (stop and wait ARQ)

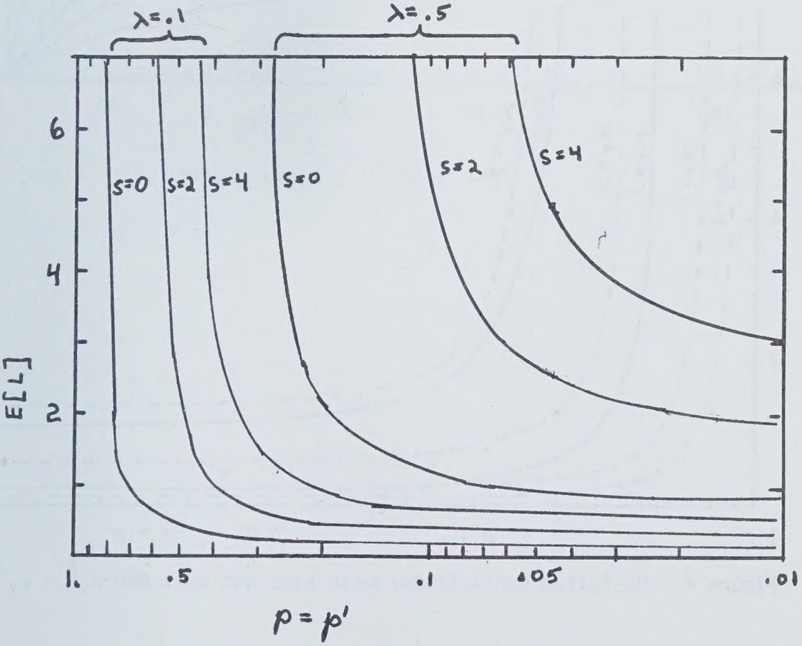


Figure 2. Average Queue Length versus p (continuous ARQ)

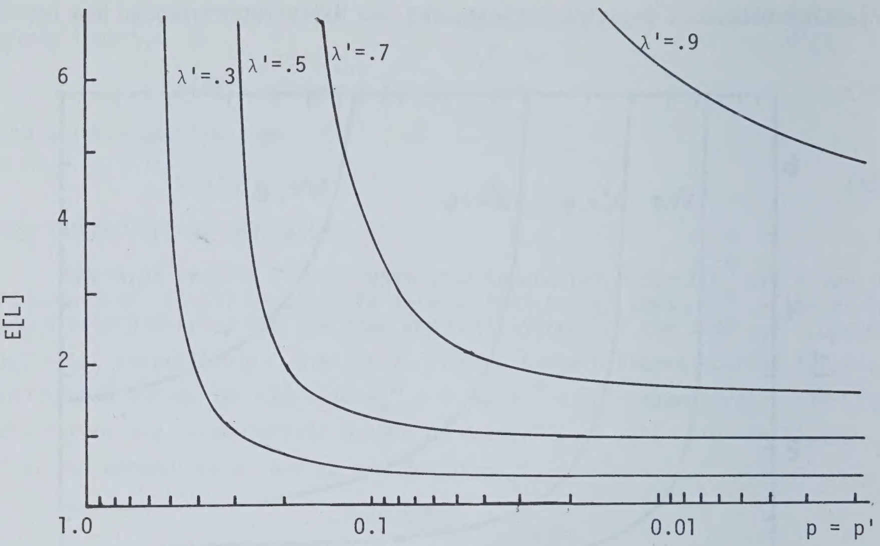


Figure 3. Unslotted Concentrator with Stop and Wait ARQ $\sigma_Y^2 = 0$

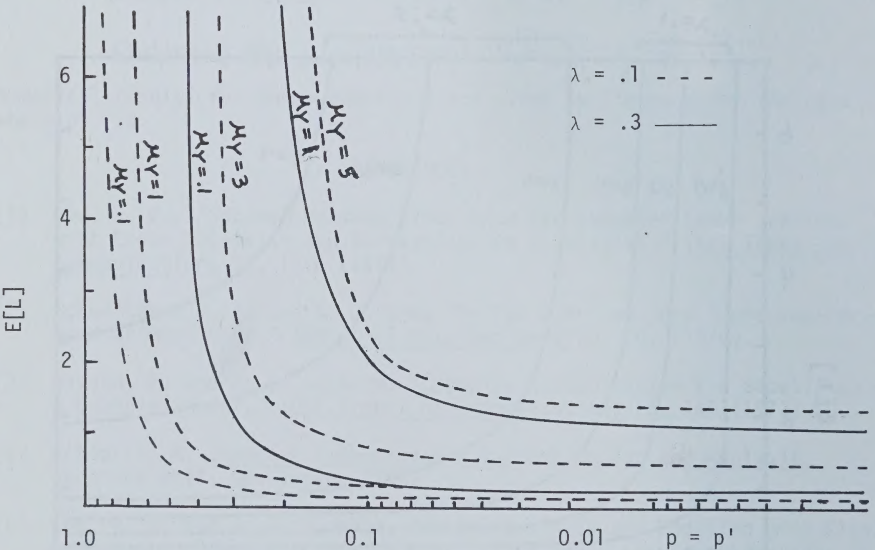


Figure 4. Unslotted Concentrator with Stop and Wait ARQ $\sigma_Y^2 = \mu_Y^2$

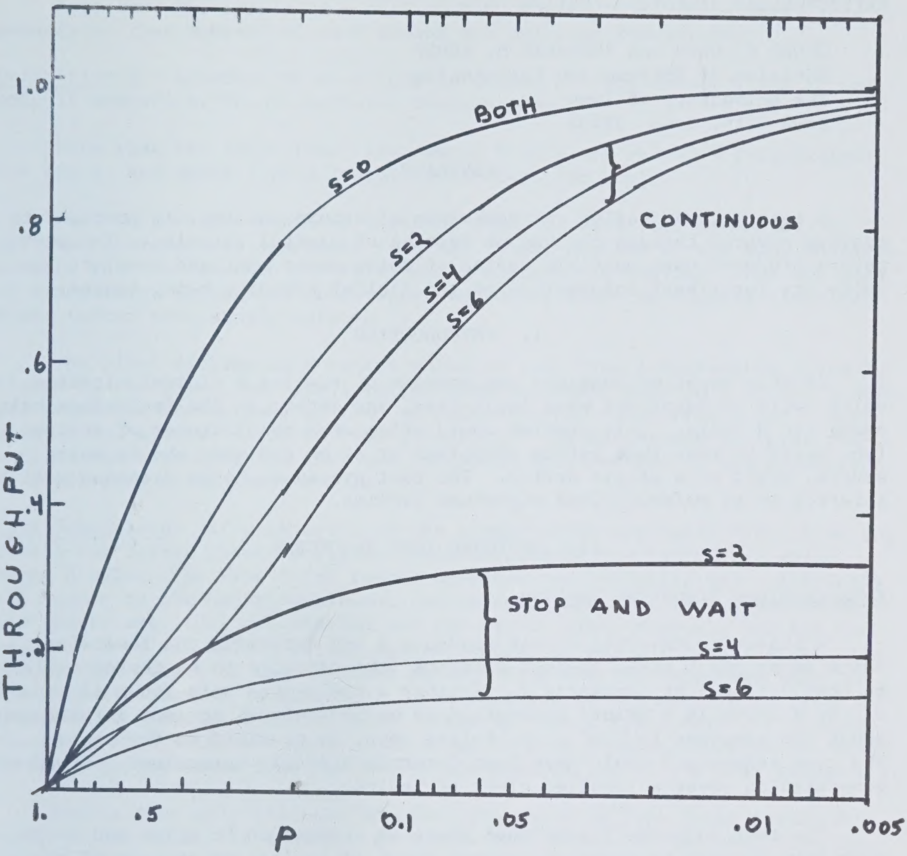


Figure 7. Throughput versus p (with p' = 0)

REFERENCELESS SIGNATURE TESTING[*]

JAMES G. QUAM and SUDHAKAR M. REDDY
Division of Information Engineering
The University of Iowa
Iowa City, Iowa 52242

ABSTRACT

A test method, called referenceless signature testing, is proposed to provide several testers for use in testing of digital circuits. The test method proposed uses multiple copies of units under test and does not require any functional information on the digital circuits being tested.

I. INTRODUCTION

In this paper we consider the problem of testing N digital circuits, which would be identical when fault-free, and determine the fault-free units among the N units. This problem could arise at a manufacturer of devices (who needs to test them before shipping) or at an end user who is using several units of a single device. The test procedure being proposed will be referred to as *referenceless signature testing*.

II. PROPOSED TEST PROCEDURE

Referenceless Signature Testing

A tester is to apply a test sequence T and determine the fault-free units among the N given digital circuits (the circuits to be tested could be combinational or sequential). Neither a fault-free unit (that is none of the N units is a priori guaranteed to be fault-free) nor any information about the response to T of a fault-free unit, is provided to the tester. The test sequence T could have been deterministically determined or randomly generated to cover a required class of faults.

The test sequence T mentioned above is assumed to be given and could have been derived to cover an assumed class of faults, by any one of the methods known earlier [1-5]. A tester performing referenceless signature testing does not have any information about the output sequence of a fault-free unit except *the knowledge that all fault-free units produce identical output sequences for identical input sequences.* This is the essential fact used by the referenceless signature testers proposed.

In the next section we show that, under reasonable assumptions, the correct units among the N units under test can be determined by using simple testers, that perform referenceless signature testing. The essential feature of these testers is that they simultaneously apply the test sequence T to more than one unit under test. We will use the following notation and terminology

T = sequence of test inputs to be applied to the units under test,
N = total number of units to be tested at one time, and
UUT = one of the N units under test.

Definition 1: Two UUTs, say X and Y, are said to be T-equivalent if and only if identical output sequences are produced by X and Y when T is applied to them.

---

[*] The research reported has been supported by NSF Grant ENG 72-04142 A02.

Note that the application of T to N UUTs partitions these units into subsets so that members of each subset are pairwise T-equivalent.

Definition 2: A subset TE of N UUTs is called a T-equivalent class if and only if members of TE are pairwise T-equivalent.

Note that the fault-free units among N UUTs are pairwise T-equivalent, for any T, and hence form a T-equivalent class for any T.

Next we give two types of testers, together with test schedules, to perform referenceless signature testing. These are two of the many possible testers that can be used for referenceless signature testing. For the sake of simplicity, throughout this paper, we assume that the digital circuits being tested have single outputs.

The block diagram of a tester which we call Type I tester, is given in Figure 1. As shown the Type I tester simultaneously applies T to all N UUTs and processes the outputs of them to determine the fault-free units. Next we give two special cases of Type I testers. The difference in these testers is the test schedule they use, which in turn depends on the proportion of fault-free units among N UUTs.

Test Schedule A: If a majority of the given N UTTs are fault-free, then the Type 1-(a) tester given in Figure 2 will determine the fault-free units among N UUTs. The Type I-(a) tester operates the following way. Initially all inputs to the majority element (whose output is 1(0) if a majority of the inputs are 1(0)) are enabled and the tester simultaneously applies the sequence T to all UUTs. If the current output of any UUT is different from the current output of the majority element then the corresponding input to the majority element is disabled. Once an input to the majority element is disabled, it remains disabled until reset to test a fresh batch of N units. The fault-free units are those UUTs whose outputs remained enabled as inputs to the majority element at the end of the application of the test sequence T.

Notice that only the test sequence generator in Type I-(a) tester depends on the UUTs. If randomly generated test sequences are used then the Type I-(a) tester can be totally independent of the UUTs and hence could be called a universal tester.

Test Schedule B: If the number of units in the T-equivalence class containing the fault-free units is larger than the number of units in any other T-equivalence class then the Type I-(b) tester shown in Figure 3 will determine the fault-free units (notice that we are now not requiring a majority of the UUTs to be fault-free). The Type I-(b) tester operates the following way. TE-i, $1 \leq i \leq N$, are N-bit registers. By storing a 1 in the jth position of TE-i, these registers indicate that the jth UUT is T-equivalent to ith UUT, $1 \leq i, j \leq N$. The AND operator shown in Figure 3 performs the bit-by-bit AND of the outputs or the complement of the outputs of all the UUTs and the contents of TE-i depending on whether the current output of the ith UUT is 1 or 0, respectively, $1 \leq i \leq N$. The result of this AND operation is stored in TE-i, $1 \leq i \leq N$. Initially TE-i's contain all ones indicating that it is assumed that before testing all UUTs are T-equivalent. As parts of the test sequence T are applied to UUTs the TE-i contains a 1 in the jth position if and only if the outputs of ith and jth UUTs have been identical up to that time. After the complete application of T, the TE-i containing the largest number of 1's will give the T-equivalence class containing the fault-free unit and hence all the fault-free units.

As an example to illustrate the Type I-(b) tester consider 6 UUTs "realizing" the combinational logic function given in Table I. Assume that the 1st, 2nd and the 3rd UUTs are fault-free and among the remaining three units two have fault flt 1 and one has fault flt 2 (the functions realized by these faulty units are given in Table I). The contents of TE-i at different time units are shown in Table II when the test sequence T = 000,001, 011,100 is applied. At the end of time unit 4 contents of the TE-1, TE-2 and TE-3 indicate that the 1st, 2nd and the 3rd UUTs are fault-free.

It can again be seen that only the test sequence generator in the Type I-(b) tester depends on the UUT.

III. AN ANALYSIS OF THE PROPOSED TEST PROCEDURE

Next we will analyze the proposed test procedures. The analysis will be divided into two sections. Section I will analyze the proposed test procedures when deterministically derived test sequences are used. Section II will analyze the proposed test procedures when randomly generated test sequences are used.

Analysis of the Proposed Test Procedure Using Deterministically Derived Test Sequences

In this analysis the following assumptions and definitions will be used.

(i) The probability that a given UUT is faulty is p.
(ii) The number of possible faults in a UUT is M.
(iii) Each one of the M faults is equally likely.
(iv) The test sequence T is simultaneously applied to N UUTs and the faults in the N UUTs are statistically independent.
(v) P_B = the probability of the occurrence of one of the M faults $= p/M$.
(vi) P_F = the probability of accepting one or more faulty units after test schedule A or B.
(vii) P_R = the probability of rejecting one or more fault-free units after test schedule A or B.
(viii) Application of a test sequence T to N UUTs will partition the N units into T-equivalent classes. ■

We will now give formulae or formulae for upper bounds on P_F and P_R for test schedules A and B. These will then be plotted for different values of N, M and p.

P_F and P_R for Test Schedule A

$$P_F = M \sum_{j=\left\lfloor \frac{N}{2} \right\rfloor+1}^{N} \binom{N}{j} P_B^j \, (1 - P_B)^{N-j}$$

$$P_R = \sum_{j=1}^{\left\lfloor \frac{N}{2} \right\rfloor} \binom{N}{j} (1 - p) \, p^{N-j}$$

P_F and P_R are plotted in Figure 4 for various values of N, M and p. P_F is usually considered to be a very significant parameter in acceptance testing. P_F can be driven quite low by increasing N. There are two possible drawbacks to a large value of N. First, if $p > 0.5$, P_R will increase with the size of N. Second, the cost of the tester may increase with N.

P_F and P_R for Test Schedule B

Formulae for upper bounds on P_F and P_R are given here without proof.

$$P_F \leq \sum_{j=0}^{\left\lfloor \frac{N}{2} \right\rfloor} \binom{N}{j} (1-p)^j \sum_{K=\begin{cases}1, & j=0\\ j, & j>0\end{cases}}^{N-j-1} \frac{1}{(N-j-K)!} \left[\sum_{m=0}^{N-j-K} \binom{N-j-K}{m} (-1)^m (N-j-K-m)^{N-j} \right] \cdot P_B^K \prod_{\ell=0}^{N-j-K-1} (p - \ell P_B)$$

$$P_R \leq \sum_{j=1}^{\left\lfloor \frac{N}{2} \right\rfloor} \binom{N}{j} (1-p)^j \sum_{K=j-1}^{N-j-1} \frac{1}{(N-j-K)!} \left[\sum_{m=0}^{N-j-K} \binom{N-j-K}{m} (-1)^m (N-j-K-m)^{N-j} \right] \cdot P_B^K \prod_{\ell=0}^{N-j-K-1} (p - \ell P_B)$$

P_F and P_R for test schedule B are plotted in Figures 5 and 6 for various values of N, M and p. P_R values for test schedule B are lower than comparable P_R values for test schedule A. However, this improvement in P_R is realized at the expense of higher P_F values for test schedule B as compared to test schedule A. In any given testing situation p and M would be fixed. The relative importance placed on P_F and P_R would dictate which test schedule to be used. As an example let $p = 0.2$ (a reasonable value for LSI devices under commercial production) and $M \geq 1000$. By choosing test schedule B and $N = 10$, $P_F < 10^{-7}$ and $P_R < 10^{-5}$. By increasing N to 20, $P_F < 10^{-13}$ and $P_R < 10^{-11}$. If schedule A were picked and $N = 10$, $P_F < 10^{-17}$ and $P_R < 5 \times 10^{-2}$. Using $N = 20$ gives the results, $P_F < 10^{-32}$ and $P_R < 5 \times 10^{-3}$.

Analysis of the Proposed Test Procedure Using Randomly Generated Test Sequences

The following amendments and additions will be made to the previous list of assumptions.

(i) UUT's are combinational circuits and faults in the UUT's do not introduce feedback.

(ii) r = the number of inputs to the units (assume one output).

(iii) All possible fault functions are assumed equally likely. Therefore, instead of M possible faults there will be $2^{2^r} - 1$ possible faults which are assumed equally likely.

(iv) S = the number of random input vectors (of length r) applied to a UUT.

(v) P_E = the probability that a unit is T-equivalent to the fault-free unit. $P_E = p \cdot (2^{2^r - S} - 1)/(2^{2^r} - 1)$. In this analysis it is assumed that $2^r - S >> 1$. Therefore, $P_E \approx p/2^S$. As a consequence the analysis is independent of r.

(vi) P_B = the probability of the occurrence of a fault in a T-equivalent class not containing the fault-free unit (note that the set of r-variable functions are divided into 2^S T-equivalent classes). $P_B = (p - P_E)/(2^S - 1)$.

P_F and P_R for Test Schedule A

$$P_F = \sum_{j=0}^{\left\lfloor \frac{N}{2} \right\rfloor} \binom{N}{j} (1-p)^j \sum_{i=\left\lfloor \frac{N}{2} \right\rfloor+1-j}^{N-j} \binom{N-j}{i} P_E^i (p-P_E)^{N-j-i}$$

$$+ \sum_{j=\left\lfloor \frac{N}{2} \right\rfloor+1}^{N-1} \binom{N}{j} (1-p)^j \sum_{i=1}^{N-j} \binom{N-j}{i} P_E^i (p-P_E)^{N-j-i}$$

$$+ (2^s-1) \sum_{j=\left\lfloor \frac{N}{2} \right\rfloor+1}^{N} \binom{N}{j} P_B^j (1-P_B)^{N-j}$$

$$P_R = \sum_{j=1}^{\left\lfloor \frac{N}{2} \right\rfloor} \binom{N}{j} (1-p)^j \sum_{i-0}^{\left\lfloor \frac{N}{2} \right\rfloor-j} \binom{N-j}{i} P_E^i (p-P_E)^{N-j-i}$$

P_R assumes the same values as in Figure 4 for deterministically derived test sequences when $s \geq 10$. This is largely the probability of not receiving a majority of fault-free units out of the N UUTs.

P_F is plotted in Figure 7. As can be seen from the graph, P_F is almost entirely independent of p and N. Therefore P_F can be lowered by simply increasing test length s.

P_F and P_R for Test Schedule B

Upper bounds for P_F and P_R are given here without proof.

$$P_F \leq \sum_{j=0}^{N-1} \binom{N}{j} (1-p)^j \sum_{i=1}^{N-j} \binom{N-j}{i} P_E^i (p-P_E)^{N-j-i}$$

$$- \sum_{j=0}^{1} (1-p)^j P_E \frac{1}{(N-2j-1)!} \left[\sum_{m=0}^{N-2j-1} (-1)^m \binom{N-2j-1}{m} (N-2j-1-m)^{N-j-1} \right]$$

$$\cdot P_B^j \prod_{\ell=0}^{N-2j-2} (p - P_E - \ell P_B) + \delta \quad , \quad \text{where } \delta \text{ is}$$

$$\delta = \sum_{j=0}^{\lfloor \frac{N}{2} \rfloor} \binom{N}{j} (1-p)^j \sum_{K=\begin{cases}1, j=0\\ j, j>0\end{cases}}^{N-j-1} \frac{1}{(N-j-k)!} \left[\sum_{m=0}^{n-j-K} (-1)^m \binom{N-j-K}{m} (N-j-K-m)^{N-j} \right]$$

$$\cdot\, P_B^K \prod_{\ell=0}^{N-j-K-1} (p - P_E - \ell P_B)$$

$$P_R \le \sum_{j=1}^{\lfloor \frac{N}{2} \rfloor} \binom{N}{j} (1-p)^j \sum_{i=0}^{\lfloor \frac{N}{2} \rfloor - j} \binom{N-j}{i} P_E^i$$

$$\sum_{K=j+i-1}^{N-j-i-1} \frac{1}{(N-j-i-K)!} \left[\sum_{m=0}^{N-j-i-K} \binom{N-j-i-K}{m} (-1)^m (N-j-i-K-m)^{N-j-i} \right]$$

$$\cdot\, P_B^K \prod_{\ell=0}^{N-j-i-K-1} (p - P_E - \ell P_B)$$

Values of P_F for type B testing are essentially identical with the values for type A testing (see Figure 7) and are not plotted again. P_R is plotted in Figure 8. This shows the improvement in P_R values for type B testing over type A. Under these assumptions type B testing with randomly generated test sequences seems particularly attractive. P_F can be driven low with very reasonable values of s, and P_R can be lowered as required by increasing N. Both of these are done largely independent of p.

REFERENCES

1. P. Agrawal and V. D. Agrawal, "Probabilistic analysis of random test generation method for irredundant combinational logic networks," IEEE Transactions on Computers, Vol. C-24, pp. 691-695, July 1975.

2. H. D. Schnurmann, E. Lindbloom and R. G. Carpenter, "The weighted random test-pattern generator," IEEE Transactions on Computers, pp. 695-701, July 1975.

3. H. Y. Chang, E. Manning and G. Metze, Fault diagnosis of digital systems, Wiley-Interscience, 1970.

4. A. D. Friendman and P. R. Menon, Fault detection in digital circuits, Prentice-Hall, 1971.

5. M. A. Breuer and A. D. Friedman, Diagnosis and reliable design of digital systems, Computer Science Press, 1976.

TABLE 1. EXAMPLE PROBLEM

ABC	True Function	Function with Fault flt 1	Function with Fault flt 2	Function with Fault flt 3	Function with Fault flt 4
000	1	1	1	0	1
001	1	0	1	1	1
010	0	0	0	0	0
011	0	0	0	0	1
100	0	0	1	0	0
101	1	1	1	1	1
110	0	0	0	0	0
111	1	1	1	1	1

TABLE 2. CONTENTS OF TE-i WITH RESPECT TO TIME FOR THE EXAMPLE PROBLEM

Time Unit	TE-1	TE-2	TE-3	TE-4	TE-5	TE-6
0	111111	111111	111111	111111	111111	111111
1	111111	111111	111111	111111	111111	111111
2	111001	111001	111001	000110	000110	111001
3	111001	111001	111001	000110	000110	111001
4	111000	111000	111000	000110	000110	000001

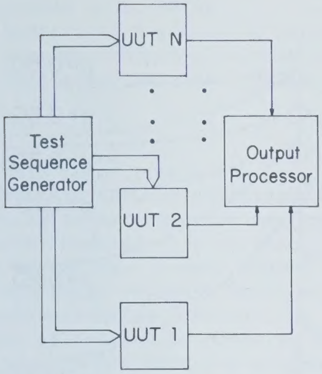

UUT N
Test
Sequence
Generator
Output
Processor
UUT 2
UUT 1

Figure 1. Type I Tester

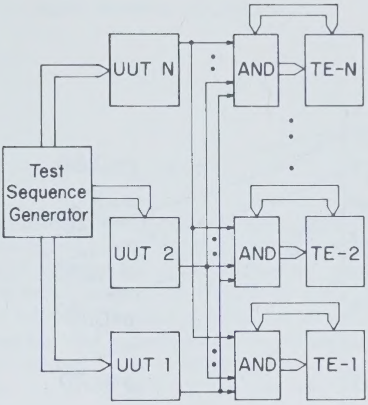

UUT N
AND
TE-N
Test
Sequence
Generator
UUT 2
AND
TE-2
UUT 1
AND
TE-1

Figure 2. Type I-(a) Tester

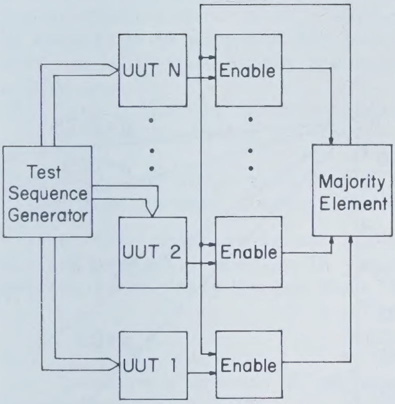

UUT N
Enable
Test
Sequence
Generator
Majority
Element
UUT 2
Enable
UUT 1
Enable

Figure 3. Type I-(b) Tester

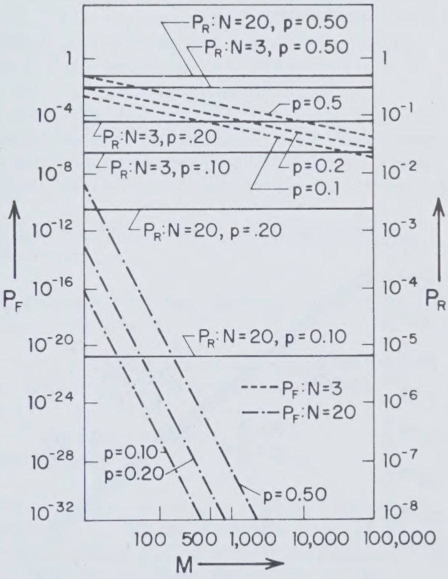


Figure 4. P_F and P_R for Test Schedule A with deterministically derived test sequences

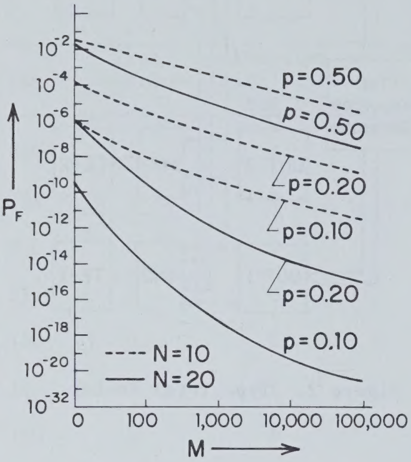


Figure 5. P_F for Test Schedule B with deterministically derived test sequences

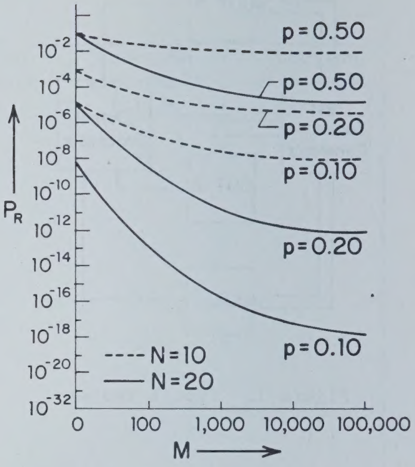


Figure 6. P_R for Test Schedule B with deterministically derived test sequences

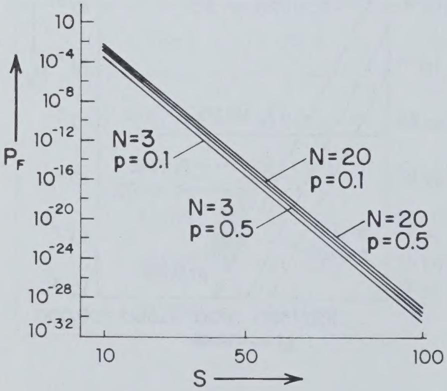


Figure 7. P_F for Test Schedule A with randomly generated test sequences

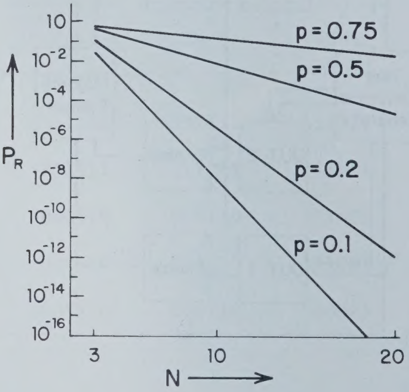


Figure 8. P_R for Test Schedule B with randomly generated test sequences

FAULT DIAGNOSIS OF MODULAR NETWORKS WITH A SMALL NUMBER OF FAULTS

GERARD G. L. MEYER
Electrical Engineering Department
The Johns Hopkins University
Baltimore, Maryland 21218

ABSTRACT

This paper treats the problems of automatic fault diagnosis of modular networks. A network consists of n modules $U_0, U_1, \ldots, U_{n-1}$ together with a testing interconnection assignment. Each module U_i is capable of testing the other modules U_j's to which it is directly connected for some specified class of faults. A very simple algorithm without backtracking is proposed to diagnose an existing fault situation among the modules, given their respective testing results. It is proved that if the number of faulty modules is smaller or equal to t, with $2t + 1 \leq n$, and if no two modules test each other, then the algorithm always gives the correct fault situation among the modules provided that either (a) a $D_{\delta,t}$ testing interconnection is used with δ and n relative prime or (b) $t \leq 3$.

INTRODUCTION

The problem of diagnosing the faulty units in a modular system consisting of n modules has been studied extensively from both the theoretical [2],[3],[6],[7] and the practical [1],[3],[5] point of view. Unfortunately, it seems that some of the fundamental properties of testing interconnections have not been discovered yet with the corollary that the algorithms proposed to resolve the diagnosis problem are unsatisfactory. For example, the algorithm proposed in [1] is based on an unproven conjecture, the algorithm proposed in [3] is incorrect according to Corluhan and Hakimi [1] and the algorithm proposed in [5] can only be used for systems in which a $D_{\delta,t}$ testing interconnection is used. In this paper, an algorithm is proposed for solving the diagnosis problem and it is shown that the algorithm yields the correct set of faulty modules in two types of situations:

(i) a $D_{\delta,t}$ testing interconnection is used with n and δ relative prime, at most t modules are faulty, and $n \geq 2t+1$.

or

(ii) every module is tested by t other modules, no two modules test each other, at most t modules are faulty, and $t \leq 3$.

At this point it is not known if the proposed algorithm always yields the set of faulty modules in situation (ii) when the number of faulty modules is strictly larger than three.

PRELIMINARIES

Consider a system S of n modules $U_0, U_1, \ldots, U_{n-1}$, and a testing interconnection set $E = \{(i,j) \mid U_i \text{ tests } U_j\}$. It is assumed that if (i,j) is in E then the test outcome a_{ij} of U_i testing U_j is $a_{ij} = 0$ if U_i believes U_j to be non faulty and $a_{ij} = 1$ if U_i believe U_j to be faulty. The diagnosis problem consists in identifying the set F of faulty modules from the knowledge of the quantities a_{ij} for all (i,j) in E. The problem is complicated by the fact that if (i,j) is in E and U_i is non faulty, then

---

This work has been supported by the Office of Naval Research under Contract NR 048 630.

U_j is non faulty when $a_{ij}=0$ and faulty when $a_{ij}=1$, but if U_i is faulty, then U_j can be non faulty or faulty regardless of the value of a_{ij}.

The approach to automatic fault diagnosis proposed in this paper is based on the use of information tables. With every module U_i is associated a table B_i with components $B_{i,0}$, $B_{i,1}, \ldots, B_{i,n-1}$. Each table is constructed by assuming that module U_i is not faulty (regardless of the fact that U_i is faulty or not). Each component $B_{i,j}$ of the table B_i contains information on the status of module U_j:

(i) $B_{i,j} = 0$ if U_i not faulty implies that U_j is not faulty,

(ii) $B_{i,j} = 1$ if U_i not faulty implies that U_j is faulty, and

(iii) $B_{i,j} = d$ if U_i not faulty has no implication on the status of U_j.

The construction of the tables B_i, $i=0,1,\ldots, n-1$ from the quantities a_{ij}, (i,j) in E is not difficult and can be done either sequentially or in parallel.

We note that if U_i is actually not faulty, then all the information contained in B_i is correct. It may happen that the construction of a table B_i yields inconsistencies: in this case, we know that module U_i is faulty. Let F_0 be the set of all modules U_i with an inconsistent table, then all modules in F_0 are faulty.

DIAGNOSIS ALGORITHM

The iterative algorithm we propose generates a monotonically increasing sequence of sets which are all contained in F, the set of faulty modules. The algorithm is based on the fact that if a module U_i is found to be faulty by more than $t+1$ modules or find more than $t+1$ modules to be faulty, it is itself faulty when it is known that the maximum number of faulty modules is t.

Given a set of modules S and the tables B_i, i not in F_0, we define scalars $c_i(S)$ and $d_i(S)$ as follows:

$$c_i(S) = \text{cardinality} \ \{j \notin S \mid B_{i,j} = 1\} \ ,$$

$$d_i(S) = \text{cardinality} \ \{j \notin S \mid B_{j,i} = 1\} \ .$$

The quantity $c_i(S)$ is simply the number of modules not in S that U_i believes to be faulty and $d_i(S)$ is the number of modules not in S which believe U_i to be faulty.

Algorithm 1: Let t be given.

Step 0: Compute the set F_0 and the information tables B_i as described in the preceding section, and set $k=0$.

Step 1: Let $F_{k+1}=F_k \cup \{i \mid c_i(F_k) \geq t+1-\|F_k\|\} \cup \{i \mid d_i(F_k) \geq t+1-\|F_k\|\}$

Step 2: If $F_{k+1} = F_k$, set $\hat{F} = F_k$, and stop; otherwise, set $k = k+1$ and to to Step 1.

The result contained in the lemma below is not difficult to prove. We note that the result does not depend on the topology of the tesing interconnection, but only on the knowledge of the maximum number of faulty modules.

Lemma 1: If $\|F\| \leq t$, then $\hat{F} \subseteq F$.

Algorithm 1 generates a lower bound for F and it is only natural to ask the following question: when is $\hat{F}$ generated by Algorithm 1 equal to F? A partial answer to this question is contained in [5] and concerns a special type of systems, namely systems in which a $D_{\delta,t}$ testing interconnection is used [6] .

Theorem 1: Let S be a system of n modules, suppose that a $D_{\delta,t}$ testing interconnection is used with δ and n relative primes, suppose that $n \geq 2t+1$, and suppose that $\|F\| \leq t$, then Algorithm 1 generates the set of faulty modules in one iteration, i.e.,

$$F_2 = F_1 = \hat{F} = F .$$

A search of t-diagnosable systems for which Algorithm 1 does not give the set of faulty modeles shows that such systems are difficult to find. This leads to the idea that, although the algorithm may not generate the complete set of faulty modules in all cases, it does generate F in many cases.

The next section of the paper shows that Algorithm 1 generates $F = \hat{F}$ when $\|F\| \leq 3$.

ANALYSIS OF THE DIAGNOSIS PROCEDURE

The analysis of Algorithm 1 consists in finding classes of modular systems for which the algorithm fails. There are no losses of generality in assuming that the set F_0 is empty. If F_0 is not empty, then we know that all the modules in F_0 are faulty and we have to identify only the remaining modules in F .

Hypothesis 1: The n modules system S and the syndrome of S satisfy:

(i) at least every module is tested by t other modules,
(ii) no two modules test each other,
(iii) t modules are faulty,
(iv) $F_0 = \emptyset$,
(v) $c_i(\emptyset) \leq t$ for $i = 0,1,\ldots, n-1$
(vi) $d_i(\emptyset) \leq t$ for $i = 0,1,\ldots, n-1$.

Clearly if Hypothesis 1 is satisfied, then Algorithm 1 will generate a set $\hat{F} = \emptyset$ which is a strict subset of F , i.e., Algorithm 1 will fail to generate the complete set of faulty modules.

Lemma 1: Assume that Hypothesis 1 is satisfied and suppose that a faulty module U_k is found to be faulty by a set M_k of non-faulty modules and a set P_k of faulty modules. Then,

(i) $p_k + m_k \leq t$,

(ii) $\|\{(i,j)\varepsilon E \mid i\varepsilon F , j\varepsilon M_k , a_{ij} = 1\}\| \geq tm_k - \frac{m_k(m_k-1)}{2} - p_k m_k$,

(iii) $\|\{(i,j)\varepsilon E \mid i\varepsilon M_k , j\varepsilon F , a_{ij} = 1\}\| \leq \frac{m_k(m_k-1)}{2}$, and

(iv) $m_k \geq 2$,

where

$m_k = \|M_k\|$, and

$p_k = \|P_k\|$.

Lemma 2: Assume that Hypothesis 1 is satisfied, then:

(i) $\|\{(i,j)\varepsilon E \mid i\not\varepsilon F, j\varepsilon F, a_{ij} = 1\}\| \geq \frac{t(t+1)}{2}$

(ii) $\|\{(i,j)\varepsilon E \mid i\varepsilon F, a_{ij} = 1\} \leq t^2$

The results of Lemma 1 and Lemma 2 enable us to characterize the modular system - syndrome combinations on which Algorithm 1 failed. Consider a system S of n modules and suppose that the t modules

$U_0, U_1, \ldots, U_{t-1}$ are faulty. Let M_k, be the set of non faulty modules which find U_k, to be faulty for $k = 0,1,2,\ldots, t-1$ and let P_k be the set of faulty modules which find U_k to be faulty for $k = 0,1,2,\ldots, t-1$.

Lemma 1 and Lemma 2 allow us to find a relationship between the cardinality of the sets $M_0, M_1, \ldots, M_{t-1}$ and t when Hypothesis 1 is satisfied.

Lemma 3: Assume that Hypothesis 1 is satisfied, then

$$\sum_{k=0}^{t-1} \frac{m_k(m_k-1)}{2} \geq \frac{t(t+1)}{2}$$

Using Lemmas 1,2 and 3 we can prove the following:

Theorem 1: There exists no syndrome of S which satisfies Hypothesis 1 when $t \leq 3$.

The proof of Theorem 1 is long but not difficult, and has been deleted.

Corollary 1: If a system S satisfies part (i), (ii), (iii) and (iv) of Hypothesis 1, and if $t \leq 3$, then at least one faulty module is found faulty by at least $t+1$ modules or find $t+1$ modules faulty.

The corollary immediately implies the main result of the paper.

Theorem 2: The set $\hat{F}$ generated by Algorithm 1 is equal to the set F of faulty modules of the system S when
(i) at least every module in S is tested by 1 other module,
(ii) no two modules test each other,
(iii) $||F|| \leq t$, and
(iv) $t \leq 3$.

CONCLUSION

The hypothesis "no two modules test each other" implies that the smallest clique in S is equal to three. It may be that the cardinality of the smallest clique in S is related to the largest t for which Algorithm 1 produces the set of faulty modules. This issue will be decided when the case $t=4$ is studied. The approach we have (although cumbersome and inelegant) should decide this issue one way or the other: either we will prove that t can be taken equal to four or we will find an example satisfying Hypothesis 1 with $t=4$.

Note that Algorithm 1 always generate the set of faulty modules when $D_{\delta,t}$ testing interconnections are used. Actually, we can generalize these results and show that Algorithm 1 always generate the set of faulty modules when $D_{\delta,t}$-like testing interconnections are used.

REFERENCES

[1] A. M. Corluhan and S. L. Hakimi, "On an algorithm for indentifying faults in a T-diagnosable system", Proceedings of the 1976 Conference on Information Sciences and Systems, The Johns Hopkins University, 1976, pp. 370-375.

[2] S. L. Hakimi and A. T. Amin, "Characterization of the connection assignment of diagnosable systems", IEEE Transactions on Computers, Vol. 23, (1974), pp. 86-88.

[3] T. Kameda, S. Toida and F. Allan, "A Diagnosing Algorithm for Networks", Information and Control, Vol. 29, (1975), pp. 141-148.

[4] S. N. Maheshwari and S. L. Hakimi, "On models of diagnosable systems and probabilistic fault diagnosis", IEEE Trans. on Computers, (1976), pp. 228-236.

[5] Gerard G. L. Meyer and Gerald M. Masson, "An Efficient Fault diagnosis algorithm for symmetric multiple processor architectures", To appear in IEEE Trans. on Computers.

[6] F. P. Preparata, G. Metze and R. T. Chien, "On the connection assignment problem of diagnosable systems", IEEE Trans. on Computers, Vol. EC-16, No. 6, (1967), pp. 848-854.

[7] J. D. Russell and C. R. Kime, "On the diagnosability of digital systems", 1973 Int'l. Symposium on Fault Tolerant Computing, IEEE Computer Society Publications, (1973), pp. 139-144.

REDUNDANCY RESULTS IN COMBINATIONAL NETWORKS BY MEANS OF L-EXPRESSIONS*

VINOD K. AGARWAL
Electrical & Computer Engineering Dept.
Wayne State University
Detroit, Michigan 48202

GERALD M. MASSON
Dept. of Electrical Engin.
The Johns Hopkins University
Baltimore, Maryland 21218

ABSTRACT

To determine a fault detection test set for a combinational switching network, it is often considered necessary to assume that every line in the network is a non-redundant line. However, for a number of reasons, most networks in practice do, indeed, contain redundant lines. Accordingly, a significant amount of research has been done [1] - [6] to study the properties of redundant lines, and their adverse effects on the capability of test sets to detect detectable faults in the presence of redundant lines.

This paper deals with previously unconsidered aspects of redundant lines by exploiting the concept of L-expressions [7] - [9]. These aspects include: (i) determining the greatest lower bound on the capability of a complete single fault detection test set for a network to cover (that is, detect) multiple faults of the network in the presence of single redundant lines in the network; and (ii) calculating an upper bound on the number of undetectable faults on multiple redundant lines.

I. Introduction:

Due to recent advances in solid state technology, it has clearly become essential to assume that the probability of occurrence of a multiple line fault in a switching network is as high as that of a single line fault. However, it is well known that generating a complete multiple fault detection test set (MFDTS) for a network is much more difficult than generating a complete single fault detection test set (SFDTS) for the same network. This difficulty mainly arises because of the large number of multiple faults which must be taken into account for generating a MFDTS. This problem is furthermore complicated if the network under consideration is redundant. This paper deals with some aspects of multiple fault testing in redundant, tree-type combinational switching networks wherein primary input fanout is allowed. The above mentioned complexity of the problem is overcome to some extent in our treatment by using the concept of an L-expression [7].

More particularly, in this paper we will determine:

(a) the greatest lower bound on the capability of SFDTS's for a network to cover (that is, detect) multiple faults of the network in the presence of single redundant lines in the network;

and (b) an upper bound on the number of undetectable multiple faults which occur on multiple redundant lines.

It will be seen that both of these bounds can be obtained by a simple table-look-up process.

II. Notational Preliminaries and L-expressions

In this section, we will first present some notational preliminaries, followed by definitions of undetectable faults and redundancy. The concept

---

* This research was supported by Office of Naval Research Contract N00014-75-C-1196

of an L-expression will next be introduced. This concept will be seen to be of central importance to the results of this paper.

Notation: (The following notation is illustrated in Figure 1.)

N: an n-input, single output, tree-type combinational switching network, wherein primary input fanout is allowed. For reference purposes, each line of N will be assumed labeled with a distinct integer.

X: $(x_1,x_2,\ldots,x_n)$, the n-tuple vector of n primary input variables to N.

X_j: $(a_1,a_2,\ldots,a_n)$, $a_i \varepsilon \{0,1\}$, $1 \leq i \leq n$, a primary input vector to N .

I,J,K: subsets of the set of lines in N .

Y_t: The Boolean function in terms of the variables of X realized by line, say, t in N . Y_t will be referred to as a line variable of N .

Y_K: $\{Y_{k_1},Y_{k_2},\ldots,Y_{k_v}\}$, $K=\{k_1,k_2,\ldots,k_v\}$. Y_I and Y_J are similarly defined.

$Z[X;Y_K]$: the primary output function of N in terms of the variables of X and, say, Y_K .

α_K: $(\alpha_{k_1},\alpha_{k_2},\ldots,\alpha_{k_v})$, $\alpha_{k_i} \varepsilon \{0,1\}$, $k_i \varepsilon K$, a binary v-tuple representing one of the 2^v possible stuck type faults which can be associated with a set K of v lines in N , wherein each of the v lines is either sa0 , or sa1 . α_K will be referred to as a *fault event* on K.

F_K: $\{\alpha_K^1 , \alpha_K^2 , \ldots , \alpha_K^{2^v}\}$, the set of all the 2^v fault events on the lines of K in N . F_K will be referred to as the *fault complex* on K .

Because of the dominating interrelationships among the faults on the lines of N , we will without any loss of generality assume that every set, such as K , of lines to be considered in the following will not contain: (a) primary input fanout stem lines; and (b) two or more lines which are on the same path from a primary input to the primary output. Because of this assumption, it follows that the primary output function $Z[X;Y_K]$ for any set K of lines of interest in N can be expressed such that each internal line variable for each $t \varepsilon K$ occurs *exactly once* in the function, either as Y_t or as $\bar{Y}_t$.

Finally, in the following, we will group together into general *coefficient terms* all parts of a primary output function, such as $Z[X;Y_K]$, consisting exclusively of primary input variables, for example, if $K=\{1,2\}$ the Boolean function $Z[X;Y_K]=(x_1\bar{x}_2Y_1+x_3) \cdot (x_4\bar{Y}_2)$ would be written as $Z[X;Y_K]=(A_1[X]Y_1+B_1[X]) \cdot A_2[X]\bar{Y}_2$, where the coefficient terms are $A_1[X]=x_1\bar{x}_2$, $B_1[X]=x_3$, and $A_2[X]=x_4$.

Redundancy: Consider a set, say, $K=\{k_1,k_2,\ldots,k_v\}$ of v lines in a network N , and a fault event $\alpha_K \varepsilon F_K$. Now, it is well known that α_K is *undetectable* if and only if $T(\alpha_K) = \phi$, where $T(\alpha_K)$ is the set of all the input vectors which can detect α_K . If a fault event α_K is detectable, then the set K is said to be a *redundant* set of lines in N . This is so because upon applying permanent binary signals α_{k_i} to each line $k_i, k_i \varepsilon K, \alpha_{k_i} \varepsilon \alpha_K$, we can remove all the lines of K from N without affecting the logical function realized by N . A set K , $|K|>1$, of lines in N is said to form a *multiple redundancy* if the set K is redundant but no proper subset of K is redundant. An undetectable fault event on a multiple redundant set of lines will be referred to as a *multiple undetectable fault event*. Various details about the concept of redundancy, and its importance in multiple fault testing problems can be found in [1] - [6]. In the following, we will assume that the reader is familiar with these references in general.

L-expressions: The concept of an L-expression has been developed [7] to generically represent the logical interconnection of line variables in a primary output function. Thus, for example, if $K=\{1,2\}$, and $Z[X;Y_K]=x_1\overline{Y}_1+x_3Y_2+x_4x_5$, then it is evident that the line variables Y_1 and Y_2 are ORed in $Z[X;Y_K]$. We represent this information using an L-expression by stating that the L-expression associated with this function which represents the logical interconnection of the lines of K within N is L+L. As pointed out before, it will be shown in the following sections that given a set K of lines in a network, useful results about the detection of fault events of F_K can be obtained simply by knowing the logical interconnection of line variables of Y_K in $Z[X;Y_K]$ (that is, simply by knowing the L-expression associated with the set K of lines).

In the following, we introduce the concept of L-expressions, and present a procedure which associates a unique L-expression with a given set K of lines. Thus, consider the following definition of L-expressions:

Definition 1: An L-expression is a combination of a given number of "L" symbols interconnected by "+" and "." operators, wherein "+" and "." are commutative and associative.

For example L+L and L·L are the only L-expressions consisting of 2 L's, and (L+L)·L is an L-expression of size 3. (Table 1 lists all the L-expressions of size 4 and less.) In the following, we will let $L(v)=\{L_v^1, L_v^2,\ldots,|L_v^p|\}$, $v\geq 1$, denote the set of all those L-expressions, L_v's, $L_v \varepsilon L(v)$, each of which consists of exactly v L's. The set $L(v)$, for any integer v, can, of course, be obtained by exhaustive considerations using Definition 1. However, a more systematic technique [7] to generate $L(v)$ has been developed, the presentation of which is beyond the scope of this paper. The interested reader is advised to refer to [7] for details. We now give a procedure which associates a unique L-expression, L_v, with a given set K, $|K|=v$, in some N.

L-expression Association Procedure:

Input: A set K of lines in N, $|K|=v$, and the primary output function $Z[X;Y_K]$.

Output: L_v, the L-expression associated with the set K of lines.

Method: Step 1: Replace every coefficient term of $Z[X;Y_K]$ by a binary 1, or a binary 0, respectively, depending on whether or not the coefficient term is ANDed with some line variable in $Z[X;Y_K]$. Simplify the remaining Boolean function by using the Boolean identities: $\hat{Y}_t+0=\hat{Y}_t$ and $\hat{Y}_t\,1=\hat{Y}_t$, where $\hat{Y}_t \varepsilon \{Y_t,\overline{Y}_t\}$, $t \varepsilon K$.

Step 2: Replace every $\hat{Y}_t$, $t \varepsilon K$, in the simplified Boolean function by the symbol "L". The resulting combination of L's is the required L_v. Stop.

Example: If $Z[X;Y_K]=(x_1\overline{x}_2Y_1+x_3)\cdot(x_4\overline{Y}_2)+x_1x_3\overline{Y}_3+Y_4+x_2x_5$ for $K=\{1,2,3,4,\}$, then $L_4=L\cdot L+L+L$.

It should be evident from the above procedure that the logical interconnection of line variable in $Z[X;Y_K]$ is uniquely represented by the L-expression L_v associated with the set K of lines. Based on this unique representation property, it is then clear that the same L-expression L_v will also be associated with any other set, say, K', $|K'|=v$, of lines in any network, say, N' if the logical interconnection of line variables in $Z'[X;Y_K']$ is the same as that of $Z[X;Y_K]$. In other words, the L-expression L_V is, in fact, a generic representation associated with an infinite number of primary output functions all of which have the same logical interconnection of line variables as that of $Z[X;Y_K]$.

Various results related to multiple fault testing in redundant combinational switching networks will now be presented in the next section by using the concept of L-expressions.

III. Redundancy Results

The problem of testing multiple faults in a switching network is usually solved by assuming that a SFDTS for the network covers "most" multiple faults as well. Recently, however, it has been shown [7] that to expediently assume that such coverage of multiple faults by a SFDTS is generally acceptable is quite tenuous, as the validity of this assumption is significantly dependent on the size of multiple fault, the network structure, and the SFDTS under consideration. It is moreover clear that if some single lines of a network are redundant, then the capability of a SFDTS for the network to cover multiple faults might be additionally affected. In this section, we will obtain the greatest lower bound on the capability of SFDTS's to cover multiple faults in the presence of single redundant lines in the network.

Another approach used to solve the problem of testing multiple faults in a network is to use a MFDTS for the network. However, this rather obvious approach is impractical because of the computational complexity involved in generating a MFDTS by using the available techniques. Moreover, most of these techniques are valid only when the network under consideration is non-redundant. In other words, to generate a MFDTS for a network, it is usually necessary to eliminate all the redundant lines (if any) from the network. In this regard, the concept of multiple redundancy is very useful [2], [5]. In this section, we will determine an upper bound on the number of multiple undetectable faults on any given set of lines. This upper bound will be useful in determining whether or not a given set of lines forms a multiple redundancy.

We start by considering the effect of single redundant lines on the multiple fault coverage capability of SFDTS's for a network. In this regard, first of all, note that it is important to have a quantitative measure of the multiple fault coverage capability of SFDTS's for a non-redundant network. As pointed out before, we have calculated such a quantitative measure in [7].

More particularly, a table, referred to as the Coverage Table, is generated in [7], wherein corresponding to each L-expression, say, L_V, $v \geq 1$, an integer, denoted as $B(L_V)$, is listed. (Some parts of the Coverage Table are reproduced here in Table 2.) It is next shown that given any set K of non-redundant lines in any network N (wherein other lines of N might, or might not be redundant) if the L-expression associated with K is L_V , then at least $B(L_V)$ fault events of f_K will be covered by each SFDTS for N . Furthermore, it is shown that there exists at least one network N , a set K of lines in N , and a SFDTS, say, T_S for N such that the L-expression associated with K is L_V and T_S covers no more than $B(L_V)$ fault events of f_K . In other words, $B(L_V)$ is the greatest lower bound on the multiple fault capability of SFDTS's to cover fault events of F_K .

It is now evident that if we determine the value of $B(L_V)$ when one or more single lines of K are redundant, then by comparison we will be able to study the effect of various single redundant lines on the capability of SFDTS's to cover fault events of F_K. As might be expected, values of $B(L_v)$ when one or more single lines of K are redundant can be expressed in a tabular form similar to the Coverage Table. In Table 3, such values of $B(L_v)$ are listed for some specific L-expressions. To conviently describe the use of such a table, we will assume the following framework.

Any set K of lines to be considered is such that $K=\{1,2,\ldots,v\}$; the line variables in $Z[X;Y_K]$ are in their uncomplemented form; and the L-expression L_v associated with K is such that counting from left to right the first "L" symbol of L_v corresponds to the line variable Y_1, the second "L" symbol of L_v corresponds to the line variable $Y_2,\ldots,$ the v^{th} and the last "L" symbol of L_v corresponds to the line variable Y_v. Moreover, we will refer to a fault event, say, $\alpha_K \varepsilon f_K$ as a $(0\rightarrow 1)$ -tested fault event if there exists at least one input vector for which when applied to N, the fault-free output of N is 0 and the faulty output of N in the presence of α_K is 1. A $(1\rightarrow 0)$-tested fault event is similarly defined. Finally, the notation k/α, $k\varepsilon\{1,2,\ldots,v\}$, $\alpha\varepsilon\{0,1\}$, will be used to denote the single fault line k saα.

To now illustrate the use of Table 3, consider a set $K=\{1,2,3,4\}$ of lines in a network N such that the L-expression associated with K is $L_4=(L\cdot L+L)\cdot L$. Now, in Columns II and VII of Table 3 we have listed, respectively, the number of $(0\rightarrow 1)$-tested and $(1\rightarrow 0)$-tested fault events of F_K which are sure to be covered by each SFDTS for N when the lines of K are non-redundant (NR). In other words, the sum of the integers in Columns II and VII is simply equal to $B(L_v)$ for NR set K. It is easily seen from this table that for $L_4=(L\cdot L+L)\cdot L$, $B(L_v)=5+11=16$. That is, all the 16 fault events of F_K will be covered by each SFDTS for N if none of the single lines of K are redundant.

Assume now, for example, that the single fault line 3 sa1 is undetectable, that is, line 3 is redundant at binary 1 value. It is then seen from Column V of Table 3 that the number of $(0\rightarrow 1)$-tested fault events of F_K which are sure to be covered by each SFDTS for N is reduced from 5 to 2. In other words, if line 3 is redundant at binary 1 value, then every SFDTS for N would cover at least $2+11=13$ fault events of F_K. Similarly, if line 4 of K is redundant at binary 0 value, then from Column XI of Table 3, we have that the value of $B(L_4)$ would be reduced to $5+6=11$. It is interesting to note from Table 3 that if line 1 or line 2 of K were redundant at binary 1 value, or if line 3 of N were redundant at binary 0 value, then the value of $B(L_4)$ will not be effected; that is, all the 16 fault events of F_K will be covered by each SFDTS for N even in the presence of any one of these three redundant lines.

Table 3 can be used in a similar manner to the above corresponding to every L-expression listed therein. For all the other L-expressions of interest, the table can be easily extended by using a procedure given in the next section. It is,however, clear from Table 3 that the effect of a single line redundancy on the multiple fault coverage capability of SFDTS's is very much dependent on the redundant lines under consideration.

It is important to note here that by using the procedure of the next section it is also possible to extend Table 3 to include those cases wherein more than one single line of K is redundant. Thus, for example, it can be shown by such an extension that for the L-expression $L_4=(L\cdot L+L)\cdot L$ considered above, if both line 4 of K were redundant at a binary 1 value and line 5 of K were redundant at a binary 0 value, then the value of $B(L_v)$ would be reduced from 16 to 8. However, if the same 2 lines were redundant at the opposite values, that is, line 4 redundant at binary 0 value and line 5 redundant at binary 1 value, then we would have that the value of $B(L_V)$ remains at 16. Here again, it is seen that the multiple fault coverage capability of SFDTS's for a network can be affected in rather extreme manners depending on the redundancy under consideration.

We now consider the use of L-expressions in studying multiple redundancy. Recall here first that a set K of lines is a multiple redundancy if and only if there exists at least one multiple undetectable fault event on the lines of K. Therefore, it should be clear that if an upper bound on the number of multiple undetectable fault events on a given set K of lines is determined, then this bound can be useful in deciding whether or not the given set K of lines forms a multiple redundancy.

As above, such an upper bound can be listed in a table, such as Table 4.

To illustrate the use of this table, consider a set $K=\{1,2,3,4\}$ of lines in a network N and assume that the L-expression, L_4, associated with K is $L_4=(L+L)\cdot(L+L)$. Now, the integer 2 listed in Column 2 of Table 4 corresponding to this L-expression implies that whatever the network N may be, there cannot be any more than 2 multiple undetectable fault events on the given set K of 4 lines. Note further from Table 4 that the only other L-expression of size 4 for which the upper bound is non-zero is $L\cdot L+L\cdot L$. It is thus easily concluded that no set of 4 lines forms a multiple redundancy in a network unless the L-expression associated with that set of lines is either $(L+L\cdot(L+L)$ or $L\cdot L+L\cdot L$. In a similar manner, it is concluded very straightforwardly from Table 4 that no set of 2 lines and 3 lines can form a multiple redundancy in a network. (Note here that this latter result has been proved elsewhere [3] in a rather involved manner.

Similar to Table 3, Table 4 can be easily extended to include all L-expressions of interest by using the procedure given in the next section.

IV. Coverage Contribution.

In this section, a brief presentation of some of the concepts and methods used in obtaining the results of the previous section is made. For details, the reader is advised to refer to [7].

Consider a set K of lines in a network N. Let X_i be an input vector to N which tests ome single fault, say, line k saα, $k \varepsilon K$, $\alpha \varepsilon \{0,1\}$, on the lines of K. Now, it is evident that X_i will also test some fault events of F_K. We will in the following refer to the exact set of fault events of F_K which are covered by X_i exclusively due to the fact that X_i tests line k saα as the coverage contribution to F_K by X_i due to k/α, where recall that k/α is used to denote line k saα. It is this concept of coverage contribution which is fundamental to the results of the last section.

Consider next a SFDTS, say, T_s for N. Clearly, $T_s \cap T(k/\alpha) \neq \phi$, for each k/α of the $2v$ single faults on the lines of K. Thus, a lower bound on the capability of T_s to cover fault events of F_K can be obtained simply by calculating the coverage contribution to F_K by each $X_i \varepsilon T(k/\alpha)$, $k \varepsilon K$, $\alpha\varepsilon\{0,1\}$, due to k/α, and by determining the minimum coverage contribution to f_K due to $2v$ single faults. However, it is obvious that calculating coverage contributions can be a quite involved task for most networks.

It can be shown, however, that lower bounds on coverage contributions to F_K can be obtained only by knowing the logical interconnection of line variables in $Z[X;Y_K]$, that is, only by knowing the L-expression associated with the set K of lines. Tables 3 and 4 are generated by using such bounds on coverage contributions. Thus, in the following, we first present an algorithm which calculates lower bounds on coverage contributions. The proof for the validity of the following algorithm will not be given here due to the lack of space; however, a complete proof can be found in [7].

The following framework will facilitate the statement of the algorithm. Let $K=\{1,2,\ldots,v\}$ be a set of v lines in N such that all the line variables in $Z[X;Y_K]$ are in the uncomplemented form. Let X_i be an input vector to N which tests a single fault, say, k/α , $k \varepsilon K$, $\alpha\varepsilon\{0,1\}$, such that the fault free output $Z[X_i;Y_K]=0$, and the faulty output in the presence of fault k/α is 1 . Then, we will use $C_K^{01}(X_i,k/\alpha)$ to denote the coverage contribution to F_K by X_i due to k/α . The superscript 01 in the notation implies that all the fault events of F_K in the set $C_K^{01}(X_i,k/\alpha)$ are $(0\to 1)$-tested fault events. The notation $C_K^{01}(X_i,k/\alpha)$ is similarly interpreted. It should be noted here that because of the above mentioned assumption on the uncomplemented form of line variables in $Z[X;Y_K]$, we would always have that a single fault k/α is $(0\to 1)$-tested ($(1\to 0)$-tested) if and only if $\alpha=1$ (correspondingly, $\alpha=0$).

In the following, we will find it convenient to consider a fault event as simply a 0-cube, and a fault complex, say, F_K , $|K|=v$, as the v-dimensional unit cube $(xx\ldots x)$. The cubical notation will also be used to represent subsets of a fault complex. Thus, for example, a set of 3 fault events, say, $\{(0,1),(1,0),(1,1)\}$ for $|K|=2$ is represented by $(1x,x1)$.

Now, in the following algorithm we will calculate $2v$ sets of cubes, denoted as, $P_k^{01}(L_v)$ and $P_k^{01}(L_v)$, $k=1,2,\ldots,v$, where L_v is the L-expression associated with the given set K of lines. These sets of cubes will be the required lower bounds on coverage contributions to F_K . More particularly, it can be shown [7] that if X_i is any input vector which $(0\to 1)$-tests k/α , $k\varepsilon K$, $\alpha\varepsilon\{0,1\}$, then there exists at least one 4-cube, say, c , $r\geq 0$, in $P_k^{01}(L_v)$ such that the fault events of F_K represented by the cube c will be contained in the coverage contribution $C_K^{01}(X_i,k/\alpha)$. Similar interpretation can be made relative to $(1\to 0)$-testing. This notation and these concepts will be made clearer after the presentation of the following lower bound coverage contribution algorithm.

Lower Bound Coverage Contribution Algorithm (LBCC Algorithm):

Input: L_v , an L-expression consisting of v L's .

Output: $P_k^{01}(L_v)$ and $P_k^{10}(L_v)$, $k=1,2,\ldots,v$.

Method: Step 1: For $v=1$, let $\Delta^\circ(L_v)=(0)$ and $\Delta'(L_v)=1$. Go to Step 5. If $v\geq 2$, go to Step 2.

Step 2: For $v\geq 2$, let $p+q=v$ such that $L_v=L_p*L_q$, where $*\varepsilon\{+,\cdot\}$. Furthermore assume that $\Delta^\circ(L_p)$, $\Delta'(L_p)$, $\Delta^\circ(L_q)$, and $\Delta'(L_q)$ are known by previous use of this algorithm. Finally, let $S_r=(\underbrace{xx\ldots x}_{r})$, $r\geq 1$.

Step 3: If $L_v=L_p+L_q$, let $\Delta^\circ(L_v)=\Delta^\circ(L_p)\times\Delta^\circ(L_q)$, and $\Delta^\circ(L_v)=\Delta'(L_p)\times S_q \cup S_p\times\Delta'(L_q)$.

Step 4: If $L_v=L_p\cdot L_q$, let $\Delta^\circ(L_p)\times S_q \cup S_p\times\Delta^\circ(L_q)$, and $\Delta(L_v)=\Delta'(L_p)\times\Delta(L_q)$.

Step 5: For $k=1,2,\ldots,v$, let $P_k^{01}(L_v)$ be the set of those cubes of $\Delta'(L_v)$ which have 1's in the k^{th} position.

Step 6: For $k=1,\ldots,v$, let $P_k^{01}(L_v)$ be the set of those cubes of $\Delta^\circ(L_v)$ which have 0's in the k^{th} position. Stop.

Now, to clarify the meaning of the sets $P_k^{01}(L_v)$'s and $P_k^{10}(L_v)$'s, consider a set $K=\{1,2,3,4\}$ of line in a network N such that the L-expression associated with K is $L_V=(L+L)(L+L)$. Furthermore, let X_i be an input vector which (0→1)-tests a single fault, say, 2/1 on the lines of K . Then, clearly, $C_K^{01}(X_i,2/1)$ represents the coverage contribution to F_K by X_i due to 2/1 . Consider now the set $P_2^{01}(L_V)$ which is, say, known, for example to be equal to = (x11x,x1x1) . It can then be shown [7] that for all X_i , the set $C_K^{01}(X_i,2/1)$ will always be such that either the fault events represented by the cube x11x , namely, {0,1,1,0),(1,1,1,0),(0,1,1,1), (1,1,1,1) ; or the fault events represented by the cube x1x1, namely, {(0,1,0,1),(0,1,1,1),(1,1,0,1),(1,1,1,1)} will be included in the set $C_K^{01}(X_i,2/1)$. In other words, a lower bound on $C_K^{01}(X_i,2/1)$ is obtained even without knowing X_i and the network N . Similar interpretations hold relative to (1→0)-testing as well.

We now present a procedure which uses sets $P_K^{01}(L_V)$'s and $P_K^{10}(L_V)$'s to calculate $B(L_V)$ for any given L-expression L_V . The value of $B(L_V)$ obtained by this procedure is for the case when each of the single lines of a set K of lines with which L_V is associated is non-redundant. The extension of this procedure to determine the value of $B(L_V)$ for redundant networks will be seen to be straightforward.

The following definition of a set of representatives will be useful in describing the procedure.

Definition 2: Given some sets of cubes, say, $c^1,c^2,\ldots,c^n$, then a set $c=(c_1,c_2,\ldots,c_r)$, $r\leq m$, of cubes is said to be a set of representative (SR) of $c^1,c^2,\ldots,c^m$ if for each set $c^i, 1\leq j\leq m$, there exists at least one cube, say, c_i in C such that $c_i \varepsilon c^j$.

Coverage Procedure:

Input: L_V , an L-expression consisting of v L's.

Output: $B(L_V)$.

Method: Step 1: Let $P_{min}^{01}(L_V)$ be a SR of $P_1^{01}(L_V),\ldots,P_V^{01}(L_V)$ such that the number of 0-cubes in $P_{min}^{01}(L_V)$ is less than or equal to the number in any other SR of $P_1^{01}(L_V),\ldots,P_V^{01}(L_V)$. Let $B^{01}(L_V)$ = number of 0-cubes in $P_{min}^{01}(L_V)$.

Step 2: Similarly, let $P_{min}^{10}(L_V)$ be a minimum sized SR of $P_1^{10}(L_V)$, $\ldots,P_V^{10}(L_V)$. Let $B^{10}(L_V)$ = number of 0-cubes in $P_{min}^{10}(L_V)$.

Step 3: $B(L_V)=B^{01}(L_V)+B^{10}(L_V)$. Stop.

It should be clear from the Coverage Procedure that $B(L_V)$ so calculated is such that if L_V is the L-expression associated with a set K of lines in some N , then every SFDTS for N will cover at least $B(L_V)$ fault events of F_K .

Now, this procedure can be easily modified to calculate the value of $B(L_V)$ when one or more single lines of K are redundant. For example, assume that line 1 is redundant at binary 0 value and line 2 is redundant at binary 1 value. Therefore, clearly, there will be no coverage contribution to F_K due to single faults 1/0 and 2/1. This observation can be taken into account in the Coverage Procedure while calculating the new value of $B(L_V)$ by choosing a set $P_{min}^{01}(L_V)$ which is a SR of only $P_1^{01}(L_V)$, $P_3^{01}(L_V)$, $\ldots,P_V^{01}(L_V)$, and by choosing a set $P_{min}^{10}(L_V)$ which is a SR of only $P_2^{10}(L_V)$, $P_3^{10}(L_V),\ldots,P_V^{10}(L_V)$. In a similar manner as above, the value of $B(L_V)$ can be calculated for any case of interest of single redundant lines of K .

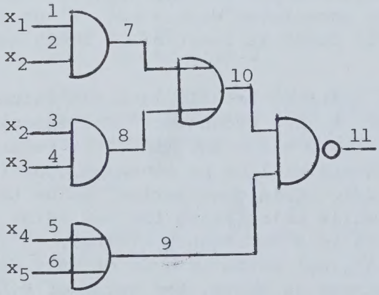
In a similar manner to the above, we can also show how this concept of coverage contribution can be used in determining an upper bound on the number of multiple undetectable faults on a given set of lines. In fact, the upper bounds discussed in the last section are very easily obtained by noting that on a multiple redundant set of lines every single line will be non-redundant. Thus, we will always have that for a given multiple redundant set K of lines at least $B(L_V)$ fault events of F_K are always covered. In other words, at the most $2^V-B(L_V)$ fault events on K can be undetectable. Thus, $2^V-B(L_V)$ is the required upper bound on the number of multiple undetectable faults on K .

V. Conclusion

In conclusion, the paper shows that the concept of L-expressions is a very useful concept for investigating various properties of redundant faults, and that the use of this concept provides some unique insights into the behavior of such faults.

REFERENCES

[1] A.D. Friedman, "Fault detection in redundant circuits", IEEE Trans. Electron. Comput., Vol. EC-16, pp.99-100 Jan. 1967.
[2] R. Dandapani and S.M. Reddy, "On the design of logic networks with redundancy and testability considerations", IEEE Trans. Comput., Vol.C-23, pp.1139-1149, Nov. 1974.
[3] H.P.S. Lee and E.S. Davidson, "Redundancy testing in combinational networks", IEEE Trans. Comput., Vol.C-23, pp.1029-1047, Oct. 1974.
[4] V.K. Agarwal and G.M. Masson, "A fault dependent Boolean difference perspective of some redundancy observations", submitted to IEEETC for review.
[5] J.E. Smith and G. Metze, "On the existence of combinational networks with arbitrary multiple redundancies", Proc. Thirteenth Annual Allerton Conf. on Circuit and System Theory, Monticello, Ill., Oct. 1975, pp.692-703; also, Report R-692, Coordinated Science Laboratory, University of Ill., Urbana, Oct.1975.
[6] V.K. Agarwal and G.M. Masson, "Resolution oriented fault interrelationships in combinational logic network", to appear in IEEE Trans.Comput.
[7] V.K. Agarwal, "Generic Characterizations of Multiple Faults for Table-Look-Up Coverage Bounding in Tree Networks". Ph.D. dissertation, Dept. Elec. Eng., The Johns Hopkins Univ., Baltimore, MD., Sept. 1977.
[8] V.K. Agarwal and G.M. Masson, "A functional form approach to fault interrelationships", Proceedings of the Fourteenth Annual Allerton Conf. on Circuit and System Theory, Univ. of Ill. at Urbana-Champaign, Urbana-Champaign, Illinois, October 1976.
[9] V.K. Agarwal and G.M. Masson, "Multiple fault coverage due to component faults," JHU Report No. 77-15.



$x = (x_1,x_2,x_3,x_4,x_5)$;

$x_i = (0,1,1,1,0)$, for example;

$I = \{1,3\}$, $J = \{5\}$, $K = \{1,3,5\}$;

$Y_9 = x_4+x_5$, for example ;

$Y_K = \{Y_1,Y_3,Y_5\}$;

$Z[X;Y_K] = (\overline{Y}_1+\overline{X}_2)(\overline{Y}_3+\overline{X}_3)+\overline{Y}_5\overline{X}_5$

$\alpha_K = (0,1,0)$, for example ;

$F_K = \{10,0,0),(0,0,1),\dots,(1,1,1)\}$.

Figure 1: Network N

v	L(v)
1	L
2	L+L,L•L
3	L+L+L,L•L+L,(L+L)•L,L•L•L
4	L+L+L+L,L•L+L+L,(L+L)•L+L, L•L•L+L,(L+L+L)•L,(L•L+L)•L, (L+L)•L•L,L•L•L•L,(L+L)•(L+L), L•L+L•L

Table 1.

L_V	B(L_V)
L	2
L+L	4
L•L	4
(L+L)•L	8
(L+L)•L•L	16
(L+L)•(L+L)	14
(L•L+L)•(L+L)	28
L+L+L+L+L+L+L	64
(L•L+L+L)•(L+L)	54
(L•L+L+L)(L•L+L)	100
((L•L+L)•L+L•L)(L+L)	206

Table 2

I	II	III	IV	V	VI	VII	VIII	IX	X	XI
Lv	NR	1/1	2/1	3/1	4/1	NR	1/0	2/0	3/0	4/0
L	1	0	-	-	-	1	0	-	-	-
L•L	1	1	1	-	-	3	2	2	-	-
L•L•L	1	1	1	1	-	7	6	6	6	-
(L+L)•L	3	2	2	3	-	5	5	5	2	-
(L+L+L)•L	9	6	6	6	9	7	7	7	7	2
(L•L+L)•L	5	5	5	2	5	11	10	10	11	6
(L+L)•L•L	3	2	2	3	3	13	13	13	10	10
L•L•L•L	1	1	1	1	1	15	14	14	14	14
(L+L)•(L+L)	9	6	6	6	6	7	7	7	7	7

Table 3.

L_V	an upper bound	L_V	an upper bound
L	0	(L+L)•L•L	0
L+L	0	L•L•L•L	0
L•L	0	(L+L)•(L+L)	2
L+L+L	0	L•L+L•L	2
L+L+L	0	(L•L+L)•(L+L)	4
(L+L)•L	0	L+L+L+L+L+L	0
L•L•L	0	(L•L+L+L)(L•L+L)	28
L+L+L+L	0	((L•L+L)•L+L•L)•(L+L)	50
L•L+L+L	0		
(L+L)•L+L	0		
L•L•L+L	0		
(L+L+L)•L	0		
(L•L+L)•L	0		

Table 4

SELF-CHECKING ALTERNATING LOGIC: COMBINATIONAL NETWORK ANALYSIS

SCOTT E. WOODARD AND GERNOT METZE
Coordinated Science Laboratory
Urbana, Illinois

ABSTRACT

This paper presents new results in self-checking alternating logic concerning the determination of the single fault self-checking property of single and multiple output self-dual irredundant combinational networks.

INTRODUCTION

Alternating logic uses time redundancy to obtain its self-checking capability. Although little work was done in this area of fault tolerant computing since Bark and Kinne's paper in 1951 [BARK], recently interest has increased with research by Yamamoto, Watanabe, and Urano in 1970 [YAMA], Reynolds and Metze in 1976 [REYN] and Shedletsky in 1977 [SHED]. This paper will discuss additional advances made.

Previous work on analyzing whether self-dual combinational networks are self-checking has centered on evaluation of the network as a whole. Yamamoto [YAMA] showed that the class of two-level networks is self-checking and Reynolds [REYN] showed that some multiple level classes are self-checking. This paper presents a technique for evaluating if any network is self-checking by providing an algorithm to evaluate the network with respect to each line.

BASIC CONCEPTS

Throughout the paper the single fault model will be used. The fault may be transient or permanent. Self-checking and alternating networks are defined as follows.

Definition 1. A self-checking network is a network which satisfies the two constraints (with input X different from input Y):

(a) $\forall f, \exists X \ni F(X) \neq F_f(X)$

(b) $\nexists f \ni \{\exists[X \in \{\text{code inputs}\} \;\&\; Y \in \{\text{code inputs}\}] \ni [F_f(X) = F(Y)]\}$

Definition 2. An alternating network is a network which for an input sequence $(X,\overline{X})$ generates an output sequence $(F(X),F(\overline{X}))$ such that $F(\overline{X})=\overline{F}(X)$.

An alternating network is an implementation of a self-dual function. Any function can be converted to a self-dual function by the addition of only one input [REYN]. Using the previous definitions it is now possible to define a self-checking alternating logic (SCAL) network.

Definition 3. An alternating network is a SCAL network iff the function realized satisfies two conditions (with input sequence $(Y,\overline{Y})$ different from input sequence $(X,\overline{X})$):

(a) $\forall f \exists X \ni (F(X),F(\overline{X})) \neq (F_f(X),F_f(\overline{X}))$

(b) $\nexists f \ni \{\exists[(X,\overline{X}) \;\&\; (Y,\overline{Y})] \ni [(F_f(X),F_f(\overline{X})) = (F(Y),F(\overline{Y}))]\}$

GENERAL SELF-CHECKING REQUIREMENTS

To simplify the presentation of results, the following symbolism will be used:

X represents an arbitrary input vector.

G(X) represents the value of an arbitrary line g for input X

F(X,G(X)) represents the network output for input X, with line g having value G(X).

$F(X,s)$ represents the network output for input X, with line g stuck-at s, $s \in (0,1)$.

For correct operation, the alternating output is represented by:

$$(F(X,G(X)),F(\overline{X},G(\overline{X}))) = (F(X,G(X)),\overline{F}(X,G(X)))$$

In order to determine whether a network is self-checking, by Definition 1 it is necessary for the network to be self-testing and fault secure. To provide a procedure to determine whether a network satisfies these conditions, it will be helpful to know whether a given line causes the network to violate one of the conditions.

Theorem 1. An alternating logic network will generate an incorrect alternating output iff there exists a line g such that:

$$[F(X,G(X)) \neq F(X,s)] \;\&\; [F(\overline{X},G(\overline{X})) = F(\overline{X},s)] = 1$$

Proof: When X is the input and line g is stuck-at s then the equation above means that the output is the opposite of what it should be in the first time period. Also, when $\overline{X}$ is applied, the output of the faulty network is the opposite of what it should be in the second time period. Let Y be the value of $F(X,G(X))$. Then $F(X,s) \neq F(X,G(X)) \Rightarrow F(X,s) = \overline{Y}$ and $F(\overline{X},G(\overline{X})) = \overline{F}(X,G(X)) = \overline{Y}$, so $F(\overline{X},s) = F(\overline{X},G(\overline{X})) \Rightarrow F(\overline{X},s) = Y$. Therefore, the output is $(\overline{Y},Y)$ an incorrect alternating output, since the output should be $(Y,\overline{Y})$.

Conversely, if $F(X,G(X)) \neq F(X,s)$, but $F(\overline{X},G(\overline{X})) = F(\overline{X},s)$, then $F(\overline{X},G(\overline{X})) = \overline{Y} \Rightarrow F(\overline{X},s) = \overline{Y}$ and the output would be $(\overline{Y},\overline{Y})$ which is a nonalternating output. Similarly when $F(X,G(X)) \neq F(\overline{X},s)$, but $F(X,G(X)) = F(X,s)$, then the output would be a nonalternating (Y,Y). Therefore only if the conditions given exist will an incorrect alternating output be generated. Q.E.D.

Using Theorem 1 tests can be derived to detect stuck-at faults on line g. The analysis proceeds as follows:

Let: $A = F(X,0) \oplus F(X,G(X))$
$B = F(\overline{X},0) \oplus F(\overline{X},G(\overline{X}))$
$C = F(X,1) \oplus F(X,G(X))$
$D = F(\overline{X},1) \oplus F(\overline{X},G(\overline{X}))$
$E = A \;\&\; B$
$F = C \;\&\; D$

Theorem 2. Using the above symbol definitions, iff $E=0$ can line g be tested for stuck-at 0 faults and iff $F=0$ can line g be tested for stuck-at 1 faults. If $E=0$ then [A V B] are tests for stuck-at 0 faults. If $F=0$ then [C V D] are tests for stuck-at 1 faults.

Proof: Let $E = 0$
then $A \;\&\; B = 0$
and $\overline{A \;\&\; B} = 1$
and $\overline{A} \vee \overline{B} = 1$
and $[F(X,0) \oplus F(X,G(X)) = 0] \vee [F(\overline{X},0) \oplus F(\overline{X},G(\overline{X})) = 0] = 1$
and $[F(X,0) = F(X,G(X))] \vee [F(\overline{X},0) = F(\overline{X},G(\overline{X}))] = 1$
Letting $F(X,G(X)) = Y \Rightarrow F(\overline{X},G(\overline{X})) = \overline{Y}$.
This implies $[F(X,0) = Y] \vee [F(\overline{X},0) = \overline{Y}] = 1$.

So the output when g is stuck-at 0 for input $(X,\overline{X})$ should be $(Y,\overline{Y})$ and when $E=0$ it is $(Y,\overline{Y})$ or (Y,Y) or $(\overline{Y},\overline{Y})$. If it is required that [A V B =1], then:

$A \vee B = 1$
and $[F(X,0) \oplus F(X,G(X)) = 1] \vee [F(\overline{X},0) \oplus F(\overline{X},G(\overline{X})) = 1] = 1$
and $[F(X,0) \neq F(X,G(X))] \vee [F(\overline{X},0) \neq F(\overline{X},G(\overline{X}))] = 1$.
Again letting $[F(X,G(X)) = Y] \Rightarrow [F(\overline{X},G(\overline{X})) = \overline{Y}]$.
This implies $[F(X,0) = \overline{Y}] \vee [F(\overline{X},0) = Y] = 1$.

So the output for g stuck-at 0 for input $(X,\overline{X})$ when [A V B = 1] is $(\overline{Y},Y)$ or $(\overline{Y},\overline{Y})$ or (Y,Y). Combining the results, when $E=0$ and inputs are applied so that $[A \vee B = 1]$, then the output for g stuck-at 0 is $(\overline{Y},\overline{Y})$ or (Y,Y). These are non-alternating outputs. This is detected and thus the fault is detected.

Conversely, assume by way of contradiction that E=1. Then the proof is:

$E = 1$

then $A \,\&\, B = 1$

and $[F(X,0) \neq F(X,G(X))] \,\&\, [F(\overline{X},0) \neq F(\overline{X},G(\overline{X}))] = 1$

Letting $F(X,G(X)) = Y \Rightarrow F(\overline{X},G(\overline{X})) = \overline{Y}$ and $[F(X,0) \neq Y] \,\&\, [\overline{F}(X,0) \neq Y] = 1$.

This implies $[F(X,0) = \overline{Y}] \,\&\, [F(\overline{X},0) = Y] = 1$.

So for input $(X,\overline{X})$ the output should be $(Y,\overline{Y})$ but when E=1, the output is $(\overline{Y},Y)$, an incorrect alternating output. Since the output alternates, the fault cannot be detected. Therefore only if E=0 can the fault be detected.

The proof for the case of F=0 is similar. Q.E.D.

A method has been given to determine whether particular inputs check whether the network is self-checking for faults on specific lines. It may turn out that there does not exist an input which checks for certain faults. When this happens the network is determined to not be self-checking.

Theorem 3. Using the previous symbol definitions, if for any line g, $[\nexists X_i \ni E = 0]$ or $[\nexists X_i \ni F = 0]$, then the network is not self-checking.

Proof: From Definition 1, to be self-checking it is required that there be a test for every potential fault. Each line may fail and so become stuck-at 0 or stuck-at 1. Theorem 2 specifies that the condition E=0 must be satisfied for line g stuck-at 0 to be tested, and the condition F=0 must be satisfied for line g stuck-at 1 to be tested. If there is no input X_i such that the condition required for a fault on line g to be tested is satisfied, then the network is not self-checking with respect to that line. Therefore the network is not self-checking. Q.E.D.

Furthermore, if a line in the network cannot be tested for either a stuck-at 0 or a stuck-at 1 fault, then the network is independent of the logic value of the line.

Theorem 4. Using the previous symbol definitions, for a given line g, if $[A \vee C = 0]$, then the line is redundant.

Proof: Let $[A \vee C = 0]$. This implies A=0 and C=0. A=0 implies $F(X_i,0)=F(X_i,G(X_i))$ for all X_i and C=0 implies $F(X_i,1)=F(X_i,G(X_i))$ for all X_i. So, $[A \vee C = 0]$ requires that the network output have the same value for either value of the line g for all inputs. Therefore the value of the line has no effect on the output and so is redundant. Q.E.D.

In addition, if a line can be tested for a fault in only one direction, say s, then the subnetwork generating the line value may be removed and replaced by a constant input of $\overline{s}$. In further analysis it will be assumed that all such replacements have been done. The fault model for the line includes only stuck-at s faults, since stuck-at $\overline{s}$ is the correct operation of the line.

It is necessary to consider the problem of a multiple line redundancy, where a combination of several lines together are redundant, but any one is not redundant by itself. Smith and Metze [SMIT] discuss this problem, but for the discussion here it is assumed that all the redundancies are single line redundancies. The redundancies will also be asssumed to be unintentional, i.e., not intended for such purposes as protecting from sequential logic hazard conditions.

Theorem 5. If a self-dual network is irredundant, then it is self-testing.

Proof: If the network is irredundant, then there does not exist any line in the network which satisfies the equation $[A \vee C = 0]$ of Theorem 4. Since , then $A=0$ implies $E=0$. Similarly $C=0$ implies $F=0$. So by Theorem 2 any line can be tested. Assuming all inputs are applied at some time, the network is self-testing. Q.E.D.

In further analysis it will be assumed the networks are irredundant, all inputs are applied, and hence the networks are self-testing. Therefore to satisfy the conditions of Definition 1 for a network to be self-checking, it will only be necessary to consider whether it is fault-secure.

To determine whether a network line prevents the network from being self-checking, the equation in Theorem 1 is evaluated. To simplify the analysis, a corollary to Theorem 1 can be derived.

The equation in Theorem 1 is:

$$[F(X,G(X)) \neq F(X,s)] \,\&\, [F(\overline{X},G(\overline{X})) \neq F(\overline{X},s)] = 1$$

By algebra, this translates to:

$$[F(X,G(X)) \,\&\, \overline{F}(X,s) \vee \overline{F}(X,G(X)) \,\&\, F(X,s)]$$
$$\&\, [F(\overline{X},G(\overline{X})) \,\&\, \overline{F}(\overline{X},s) \vee \overline{F}(\overline{X},G(\overline{X})) \,\&\, F(\overline{X},s)] = 1 .$$

Since alternating logic is used, by Definition 2,

$$F(\overline{X}) = \overline{F}(X) \quad \text{or} \quad F(\overline{X},G(\overline{X})) = \overline{F}(X,G(X))$$

Using this relation, the above equation changes to:

$$[F(X,G(X)) \,\&\, \overline{F}(X,s) \vee \overline{F}(X,G(X) \,\&\, F(X,s)]$$
$$\&\, [F(X,G(\overline{X})) \,\&\, \overline{F}(\overline{X},s) \vee F(X,G(X)) \,\&\, F(\overline{X},s)] = 1$$

By algebra, this reduces to:

$$F(X,G(X)) \,\&\, \overline{F}(X,s) \,\&\, F(\overline{X},s) \vee \overline{F}(X,G(X)) \,\&\, \overline{F}(\overline{X},s) \,\&\, F(X,s) = 1 .$$

Since alternating logic is used, whenever X_i is applied, $\overline{X}_i$ is also applied in the alternating pair $(X_i,\overline{X}_i)$. So if for some alternating pair $(X_i,\overline{X}_i)$ for input X_i, $\overline{F}(X_i,G(X_i)) = 1$ and $F(X_i,s) = 1$ and $\overline{F}(\overline{X}_i,s) = 1$

then for another alternating pair $(X_j,\overline{X}_j)$ with $X_j=\overline{X}_i$ and $\overline{X}_j=X_i$

$\overline{F}(\overline{X}_j,G(\overline{X}_j)) = F(X_j,G(X_j)) = 1$ and $F(\overline{X}_j,s) = 1$ and $\overline{F}(X_j,s) = 1$

So if all input pairs are applied, then if for some X

$$\overline{F}(X,G(X)) \,\&\, F(X,s) \,\&\, \overline{F}(\overline{X},s) = 1$$

then also for another X

$$F(X,G(X)) \,\&\, \overline{F}(X,s) \,\&\, F(\overline{X},s) = 1 .$$

So only one of the above products needs to be checked to determine the self-checking characteristics. To analyze for both stuck-at 0 and stuck-at 1 faults, s is replaced by the appropriate value as the following corollary of Theorem 1 states.

Corollary 1. The network is self-checking with respect to a line g iff line g is in an irredundant self-dual network and satisfies

$$F(X,G(X)) \,\&\, [\overline{F}(X,0) \,\&\, F(\overline{X},0) \vee \overline{F}(X,1) \,\&\, F(\overline{X},1)] = 0 .$$

Corollary 1 may be broken down to consider separately stuck-at 0 and stuck-at 1 faults in line g. The network is self-checking with respect to stuck-at 0 faults in line g iff

$$F(X,G(X)) \,\&\, \overline{F}(X,0) \,\&\, F(\overline{X},0) = 0 .$$

The network is self-checking with respect to stuck-at 1 faults in line g iff

$$F(X,G(X)) \,\&\, \overline{F}(X,1) \,\&\, F(\overline{X},1) = 1 .$$

The equation in the corollary may also be written in different forms taking advantage of the equivalences stated before. These revised equations are:

$$F(X,G(X)) \,\&\, \overline{F}(X,0) \,\&\, F(\overline{X},0) \vee F(\overline{X},G(\overline{X})) \,\&\, F(X,1) \,\&\, \overline{F}(\overline{X},1) = 0$$
$$F(\overline{X},G(\overline{X})) \,\&\, F(X,0) \,\&\, \overline{F}(\overline{X},0) \vee F(X,G(X)) \,\&\, \overline{F}(X,1) \,\&\, F(\overline{X},1) = 0$$
$$F(\overline{X},G(\overline{X})) \,\&\, [F(X,0) \,\&\, \overline{F}(\overline{X},0) \vee F(X,1) \,\&\, \overline{F}(\overline{X},1)] = 0$$

In some cases it may be simpler to evaluate these equations rather than the one given in Corollary 1.

SUFFICIENT SELF-CHECKING CONDITIONS

Rather than having to examine every line in a network to determine whether the conditions of Corollary 1 hold, it is possible to determine and use some general rules for deciding whether certain types of lines always satisfy the conditions of Corollary 1.

Theorem 6. If $\forall X, G(X) \neq G(\overline{X})$, then the network is self-checking with respect to line g.
Proof: Consider an input X_i with $G(X_i) = d$, $d \in (0,1)$. Then by the assumption in Theorem 6, $G(\overline{X}_i)=\overline{d}$. So $F(\overline{X}_i,G(X_i))=F(X_i,d)$ and $F(\overline{X}_i,G(\overline{X}_i))=F(\overline{X}_i,\overline{d})$. Now let $F(X_i,\overline{d})=Y$ and $F(X_i,\overline{d})=\overline{Y}$. If g is stuck-at s, then $G(X_i)=s$ and $G(\overline{X}_i)=s$. By Theorem 1, the network will not be self-checking iff $[F(X,G(X)) \neq F(X,s)]\ \&\ [F(\overline{X},G(\overline{X})) \neq F(\overline{X},s)] = 1$

By substitution of the values above, the equation changes to $[Y \neq F(X,s)]\ \&\ [\overline{Y} \neq F(\overline{X},s)] = 1$. Either $s=d$ or $s=\overline{d}$; without loss of generality, let $s=d$.So, $F(X,s)=F(X,d)=Y$ and the equation of Theorem 1 is not satisfied. Therefore, an irredundant self-dual network is self-checking with respect to line g. Q.E.D.

By this theorem, a network is self-checking with respect to all lines which alternate in the two time periods of alternating logic. Using Theorem 6, the result presented by Reynolds [REYN] that a network is self-checking with respect to all input lines can be easily proven since all input lines have alternating values.

If a line does not fan out, then the analysis to determine if the network is self-checking with respect to the line is simpler than if it did fan out.

Theorem 7. If a line g in a self-dual network does not fan out in its path to the output and the gates in the path are all unate, then the network is self-checking with respect to line g.
Proof: For some input X_i the line g must be sensitized if the network is irredundant. This will force the output to be the opposite of what it would otherwise be, $F(X_i,s)=\overline{F}(X_i,G(X_i))$. Since the network elements on the path from line g to the output are all monotonic and there is only one such path to the output, g can only affect the output in one direction when it is stuck-at s. Therefore, line g is not sensitized when $\overline{X}_i$ is applied. If line g is not sensitized, then the output would be unchanged and a nonalternating output pair $(\overline{F}(X_i,G(X_i)),F(\overline{X}_i,G(\overline{X}_i))) = (\overline{F}(X_i,G(X_i)),\overline{F}(X_i,G(X_i)))$ is obtained. Therefore, the failure is detected and the network is self-checking with respect to line g. Q.E.D.

If network elements that were not monotonic were allowed on the path from line g to the output, then Theorem 7 would not apply. For example, consider an exclusive-or gate on the path from line g to the output. If the output of g for $(X_i,\overline{X}_i)$ is (0,0), then when g is stuck-at 1 the output of g changes to (1,1). If the output of Q for input $(X_i,\overline{X}_i)$ is (0,1), then normally the output of f is (0,1). However when g is stuck-at 1, then the output is (1,0) and the failure of line g is undetected. In this case the network is not self-checking with respect to line g stuck-at 1.

From Theorem 7 the result of Yamamoto, Watanabe, and Urano [YAMA] that two-level self-dual networks with monotonic gates are self-checking can be easily proven. Since none of the non-input lines fan out in the single output network, the conditions of Theorem 7 are satisfied by the non-input lines. The input lines alternate and so the network is self-checking with respect to all the lines in a two level network. A level of inverters on the inputs preserves this property since their outputs alternate. Furthermore, the analysis can be applied to each output of a multiple output network. If each output is checked for alternation, then the multiple output network is self-checking.

The following result is based on the work of Reyolds [REYN].

Definition 4. Path parity from line g to the network output is the modulo 2 number of inversions on the path.

Theorem 8. If all paths from line g to the network output have the same parity, then the network is self-checking with respect to line g.

In multiple level networks the input is often fed to a level other than the first. This enhances the self-checking properties of an irredundant network as the following theorem proves. This theorem applies to a restricted set of gate types defined as follows:

Definition 5. A standard gate is a NOT,NAND,AND,NOR, or OR gate.

Theorem 9. In an irredundant network, if a line g is the input to the same standard gate as an alternating line, then the network is self-checking with respect to line g.

Proof: Since the network is irredundant and all inputs are presumed to be applied, then the network is self-testing with respect to all the lines, including line g, by Theorem 5. Now consider the fault secure property. In a standard gate one input value dominates by forcing the output to a specific value when the dominant input value is applied. This is observed by considering their truth tables. The dominant input values are 0 for NAND and AND, and 1 for NOR and OR. Since an alternating value is applied to one input line of the standard gate, then for the time period the dominant alternating input is applied, the output value is independent of the other inputs to the gate. Even if one of the other inputs is stuck-at s, the output of the gate will be correct. Therefore, the network output will be correct for that time period. During the other time period, if the network output is correct, then the fault does not affect the output. If the network output is incorrect, then it is nonalternating and the fault is detected. Either way, the network is fault secure. Since the network is both fault secure and self-testing, by Definition 2.4 it is self-checking. Q.E.D.

Note that Theorem 9 would not apply to a gate which is not a standard gate, such as the XOR gate or majority gate since these do not exhibit the dominance property.

Using the results presented it is now possible to follow a procedure to determine whether a self-dual combinational single-output network is a SCAL network. If a line passes any of the following tests, then an irredundant self-dual network is self-checking with respect to that line:

1. It alternates in value for all X_i.
2. It does not fan out and its path to the output is through unate gates.
3. Path parity is the same for all paths from the line to the network output.
4. It is the input to the same standard gate as an alternating line.
5. Conditions of Corollary 1 are met.

MULTIPLE OUTPUT COMBINATIONAL SCAL

So far only single output networks have been considered. In this section a procedure is presented for determining whether a multiple output network is self-checking with respect to a line in the network.

In general, multiple output networks may be classified as being of two types: separable and nonseparable. As is implied by the names, the two types differ according to whether the combinational logic generating the outputs from the alternating inputs may be entirely separated or not. If the combinational logic may be separated, then the analysis of preceding sections may be applied to the logic generating each output individually to determine whether the network is self-checking. Similarly when the outputs are not independent but a portion of their logic is, then the methods of preceding sections may be applied to the portion of the logic that is independent and thus its effect on the self-checking property of the network may be assessed.

However, when some logic is shared to generate a line common to more than one output, then the analysis procedure must be changed.

Definition 6. A multiple output network is self-checking with respect to a line shared commonly between more than one output iff for any input X_i for which an output alternates incorrectly, some other output does not alternate for that input.

To evaluate whether a line commonly shared between more than one output satisfies the self-checking requirements, a determination of the specific requirements on the line is useful. The symbol definitions used in the previous sections will be used with a subscript added to indicate a specific output line.

Theorem 10. For a line g stuck-at s, if an input X_i causes $F_j(X_i,s)=\overline{F}_j(X_i,G(X_i))$ and $F_j(\overline{X}_i,s)=\overline{F}_j(\overline{X}_i,G(\overline{X}_i))$ then some output F_k must satisfy $F_k(X_i,s)=F_k(\overline{X}_i,s)$ for the network to be self-checking.

Proof: By Definition 6 an incorrect alternating output on one network output, F_j , must be accompanied by a nonalternating output on another network output, F_k, if the network is to be self-checking with respect to a fault on line g. Q.E.D.

If the network is shown by Theorem 10 to be self-checking with respect to line g, it may still no7'e self-checking with respect to lines which fan out from g. These two lines must be checked as all other lines are checked which are used to generate only one network output.

From Theorem 10, it is observed that the requirements for lines which feed more than one network output are less strict than for other network lines which feed only one network output.

In the analysis, all lines in the combinational logic block generating a line g which fans out to more than one output are checked under the reduced requirements for multiple outputs. To evaluate whether the network is self-checking with respect to a line g shared by more than one output, first a single output is checked to determine whether it is self-checking with respect to line g. If all outputs which use line g are self-checking with respect to line g, then the network is self-checking with respect to line g by Definition 6. However, if one output is not self-checking with respect to line g, the network may still be self-checking with respect to line g according to Theorem 10. To provide a specific analysis procedure to check if this is so, Corollary 2 is derived.

Consider a stuck-at fault, s, on line g, which causes an incorrect alternating output on output F_1. The equation of Corollary 1 is then

$$F_1(X,G(X)) \;\&\; \overline{F}_1(X,s) \;\&\; F_1(X,s) \neq 0$$

If another output, F_k, has a nonalternating output for this fault then

$$F_k(X,s) = F_k(\overline{X},s) \quad \text{or} \quad [\overline{F}_k(X,s) \;\&\; \overline{F}_k(\overline{X},s) \vee F_k(X,s) \;\&\; F_k(\overline{X},s)] = 1 \;.$$

If the value(s) of X_i for which the incorrect alternating output is generated on F_1 are included among the outputs of F_k which are nonalternating, then by Theorem 10 the network is self-checking with respect to g. Furthermore, if there are several outputs which share line g, then all the inputs for which they are nonalternating are protected from not being self-checking on output 1. This is stated more precisely in Corollary 2:

Corollary 2. If $F_1(X,G(X)) \;\&\; \overline{F}_1(X,s) \;\&\; F_1(\overline{X},s) \neq 0$ for some output, but

$$\overline{\left[\sum_{k=1}^{n} \overline{F}_k(X,s) \;\&\; \overline{F}_k(\overline{X},s) \vee F_k(X,s) \;\&\; F_k(\overline{X},s\right]} \;\&\; F_1(X,G(X)) \;\&\; \overline{F}_1(X,s) \;\&\; F_1(\overline{X},s) = 0$$

then the n-output irredundant self-dual network is self-checking with respect to line g by checking all the outputs.

In the analysis of networks to determine whether they are self-checking with respect to a line g shared commonly by more than one output, the single output analysis summarized previously will be done first since it is simplest. However if g does not satisfy at least one of these conditions, then it will be examined to determine whether it passes the more relaxed requirements for multiple output network lines.

SELF-CHECKING DESIGN AND ANALYSIS ALGORITHM

The results of previous sections now provide a complete algorithm for determining whether an irredundant self-dual single or multiple-output network is self-checking. The algorithm proceeeds as follows:

Algorithm 1.

1. Each network output will be regarded as independent of the others. Each line which is used to generate an output will be examined to determine whether it satisfies at least one of the following conditions:

   A. It alternates for every input pair $(X_i,\overline{X}_i)$.

   B. It does not fan out and its path to the output is through unate gates.

   C. Path parity is the same for all paths from the line to the network output.

   D. It is the input to the same standard gate (NAND, NOR, OR, AND, or NOT) as an alternating line.

   E. It meets the condition

   $$F(X,G(X)) \;\&\; (\overline{F}(X,0) \;\&\; F(\overline{X},0) \vee \overline{F}(X,1) \;\&\; F(\overline{X},1)) = 0 \;.$$

2. If any line from a subnetwork used by more than one output fails to meet one of these conditions for an output 1, then it is checked to see if

$$\overline{\left[\sum_{k=1}^{n} \overline{F}_k(X,s) \;\&\; \overline{F}_k(\overline{X},s) \vee F_k(X,s) \;\&\; F_k(\overline{X},s\right]} \;\&\; F_1(X,G(X)) \;\&\; \overline{F}_1(X,s) \;\&\; F_1(\overline{X},s) = 0$$

3. If the line does not meet at least one of the above conditions, then the network is not self-checking.

DESIGN RECOMMENDATIONS

From the presentation of determining whether a network is self-checking, a few design recommendations surface: (1) minimize fan out, particularly of unequal parity to the output, (2) use two levels (plus an inverter level) to automatically achieve self-checking, and (3) share logic between as many outputs as possible, since this reduces the requirements on the lines for the network to be self-checking.

REFERENCES

[BARK] Bark, A., and Kinne, C., "The Application of Pulse Position Modulation to Digital Computers," _Proc. National Electronics Conference_, pp. 656-664, September 1953.

[REYN] Reynolds, D., and Metze, G., "Fault Detection Capabilities of Alternating Logic," _Proc. 6th Annual Symposium on Fault Tolerant Computing_, pp. 157-162, June 1976.

[SHED] Shedletsky, J., "A Rollback Interval For Networks with an Imperfect Self-Checking Property," Technical Report Number 96, Center For Reliable Computing, Stanford University, December 1975.

[SMIT] Smith, J., and Metze, G., "On the Existence of Combinational Networks with Arbitrary Multiple Redundancies," _Proc. of the 13th Allerton Conference_, October 1-3, 1975.

[YAMA] Yamamoto, H., Watanabe, T., and Urano, Y., "Alternating Logic and Its Application to Fault Detection," _Proc. 1970 IEEE International Computing Group Conference_, Washington, D.C., pp. 220-228, June 1970.

THE DESIGN OF TOTALLY SELF-CHECKING ARITHMETIC CIRCUITS

JEAN DUSSAULT and GERNOT METZE
Coordinated Science Laboratory
Urbana, Illinois

ABSTRACT

This paper discusses the design of totally self-checking arithmetic circuits using arithmetic codes. Arithmetic structures as well as checkers are studied. It is shown that the resulting structures are essentially the same as those for conventional checkers provided that some basic rules are followed.

1. INTRODUCTION

In this paper we shall be concerned with the checking of arithmetic operations, using well-known arithmetic codes: parity, AN and residue codes. Fixed-weight codes, like the biquinary and 2-out-of-5 codes, have been used in decimal adders and extension to general adders is far from being simple. Ho has studied a TSC two-rail adder and incorporated it in a tSC system [5].

Arithmetic codes are designed to correct or detect arithmetic errors i.e. errors that can cause borrow and carry propagation of logic faults resulting in several faulty output bits. These errors are characterized by output changes of the form $\pm 2^i \pm 2^j \pm \ldots$. Techniques to check parity of words are simple and well-known. However checking arithmetic using parity codes requires access to internal carries of the adder and duplication of carries is required for increased error coverage. As expected complexity of checking increases with increases in concurrency and look-ahead. On the other hand AN and residue checking requires no major modification to the adder. More complex adders may introduce more redundancy in the operands but the design procedure is always straightforward.

Kolupaev presented self-testing residue trees [6]. His approach combined fixed-weight code and residue codes in the sense that trees were mapping residue codes into codes of the form 1-out-of-2^n where n was the byte size. Although modular (2 types of module were used) this approach exhibits very large fan-in and is generally very expensive as well. Our approach shall be more conventional.

2. ARITHMETIC CODES

This section presents a brief overview of arithmetic codes. Parity codes will also be dealt with, but need little introduction.

Definition 1: An expression for an integer I of the form $I = \sum_{j=0}^{n-1} i_j \cdot 2^j$, $i_j \in \{-1,0,+1\}$ is called the modified binary form (MBF) of I.

Definition 2: The arithmetic weight of the integer I, denoted $W_A(I)$, is the least number of non-zero coefficients which are necessary in a MBF of I.

Definition 3: The arithmetic distance between integers I_1 and I_2, denoted $D_A(I_1,I_2)$ is the arithmetic weight of their difference i.e.

$D_A(I_1,I_2) = W_A(I_1-I_2)$.

Definition 4: An AN code can be defined as a code generated by a positive integer A and is the set of integers AN for $0 \leq N \leq B$, where B is specified.

It can easily be verified that the sum and product of AN codepoints are also AN codepoints. The code capabilities of AN codes are defined in the following theorem:

Theorem 1: For any $t \geq 0$ and $s \geq 0$, an AN code can correct all errors of weight t or less in its codepoints and can detect all other errors of weight t + s or less in its codepoints if and only if $D_{Amin} > 2t + s$.

Definition 5: Let Z_m be the set of integers modulo m. A residue code in Z_m, generated by the integer m such that $m > 0$ is the (k+1)-tuple of integers $N = \{N, [N]_m\}$ where $[N]_m = N \bmod m$. Again it can easily be seen that the check digits of a sum or a product of 2 encoded numbers are the sum or product of the check digits of the encoded numbers. These codes are also clearly systematic.

Definition 6: The associated AN code for a residue code is the AN code whose generator is $A = m$.
The code capabilities can now be defined in the following theorem:

Theorem 2: A residue code has the same error detecting and correcting abilities as its associated AN codes (Proof in [7]).

A final comment is in order. Although minimum arithmetic distance is used to define code capabilities in an arithmetic environment, codes with similar minimum distance may have completely different overall error coverage. For example the residue 3 and 15 codes will both detect all single arithmetic errors but while only 2/3 of all possible error patterns (in a 4-bit byte) are detected by the residue 3 code, the residue 15 code will detect 14/15 of all possible error patterns. This will be discussed further in section 3.

3. COMMENTS ON THE CHOICE OF A MODULUS

In theory any modulus or any generator can be used to construct arithmetic codes. In fact moduli of the form $\pm 2^i \pm 2^j \pm 2^k$..(like 11, 13, 19, 21 etc.) can be attractive since they have a minimum arithmetic distance equal to 3, hereby providing double arithmetic error detection or single arithmetic error correction. However it is relatively difficult to check those codes as they are awkward to check using regular binary arithmetic. One has either to provide additional logic to conventional adders or to design a new adder with the proper modulus so they can be used as checkers. In any case the resulting checker will have very little structure since the weight distribution of the bits in a binary encoded argument, modulo a relatively prime modulus, has an order equal to the modulus minus one. In other words the size of the bytes equals the modulus-1, which is inconvenient as usual word sizes are multiples of 8 or 12. These difficulties have prevented generalized use of such moduli.

The most common moduli are of the form 2^n-1. Although these moduli provide only single arithmetic error detection (their "real" capabilities are discussed in the next section) checking is accomplished by simple one's complement addition. Moreover the one's complement algorithm applies directly to left and right shiftings, complementation and other more specialized operations. Encodings and decoding into/from non-separate codes is also easy. Avizienis discusses the properties and details of implementation of such codes [2]. The design of checkers for (2^n-1)-encoded arguments is straightforward but it results in slow operation due to end-around carries. So far it is the only practical approach that has been used for arithmetic codes.

4. DIFFERENT CONCEPTS OF DISTANCE

The capabilities of arithmetic codes are defined in terms of their arithmetic or modular distances. On the other hand the inputs to a checker have code capabilities which are usually best measured in terms of Hamming distances. In this section we will look at some of the considerations imposed by this discrepancy.

Definition 7: The circuit distances of a circuit is the set of Hamming weights of possible errors at the circuit outputs caused by a modeled fault for code space inputs. The bound of interest in the case of circuit distance is the upper one: the maximum circuit distance.

As indicated previously the relation between arithmetic distance and Hamming distance (which is in turn simply related to circuit distance) is not straightforward: the Hamming distance in arithmetic codes is a function of the codewords themselves as well as the set of arithmetic distances. The Hamming distances between arithmetic codewords vary much more widely than the Hamming distances of good (well-packed) communication codes. However bounds on both types of distance are simply related and a sufficient conditions can be derived from those to design totally self-checking arithmetic checkers. In the following we investigate briefly this relationship and shall provide guidelines to design functional circuits that have maximum circuit distance sufficiently small for given arithmetic codes.

Theorem 3: Arithmetic codes (AN or Residue) have a minimum Hamming distance equal to the arithmetic weight of the generator or residue. (Proof in [7]).

This signifies that the minimum Hamming distance is quite small in general and therefore fault secureness could be very vulnerable to larger circuit-distances. Consider the following:

Suppose we have residue 2^n-1 and $(2^n-1)N$ codes. As n increases the redundancy increases. However, according to the above theorem, no matter how large n is, codes of the form 2^n-1 are equally vulnerable. Since the theorem states that there is a double arithmetic fault, namely 2^n-1 itself, that will not be detected. In practice, however, one has to look hard to find a single fault in the adder circuitry that produces such a double fault (for n large). The overall code capabilities actually increase and it may be justified since 2^n-2 error patterns are now detectable. Indeed the picture is not so bleak when the distribution of Hamming weight changes, at the output, is considered. This is formalized in the concept of arithmetic circuit distance set.

Definition 8: The arithmetic circuit distance set is the set of arithmetic weights of potential error patterns caused at the output be members of the fault set.

Of special concern are those patterns that can be generated at the end of paths that originated from fanout points. They have the largest arithmetic weights and are the most likely to produce changes at the outputs that are equal to the modulus or the generator.

Theorem 4: In an arithmetic circuit if the arithmetic weight of the generator or the modulus is not a member of the arithmetic circuit distance set, then the circuit is fault secure.

5. ARITHMETIC CODES AS TEST SETS

In this section we show that common arithmetic codes provide a sufficient test set for adders that are fault secure with respect to the types of arithmetic errors that the codes are designed to protect against. It will first be indicated that the AN codes considered are cyclic, that there exists a test set for a small number of consecutive full-adders and finally show that it covers all adders in a parallel adder using the cycling property.

Theorem 5: The generator A and the number of codewords B in a cyclic AN-code satisfy $AB = 2^j-1$. Conversely, every A which divides 2^j-1 generates a cyclic code with $B = (2^j-1)/A$ codewords (statement and proof in [7]).

This theorem seems to impose a constraint on the size of the code. Previously, the only requirement was that the generator be of the form 2^n-1. These two apparently conflicting cases can be reconciled by noting that the constraint $AB = 2^j-1$ simply implies that the codeword size be an integer multiple of the byte size. For example, remarking that $3 = 2^2-1$ divides 2^j-1 if and only if j is even, it can be seen that codeword sizes should also be even, i.e. of length 2,4,6 etc. Similar consideration can be applied to other moduli and is formalized in the theory of exponents.

To test an irredundant adder it is sufficient to apply all possible input combinations. Because of carry dependencies, more than one adder stage has to be considered. For ripple-through carry adders, 2 stages are sufficient; it may be more for adders with look-ahead. Because of the cycling property, this test set will be eventually applied to all stages. For 2 stages, assuming bytes of size n (i.e. generator = 2^n-1) all the combinations are produced by the first 4 codewords. Underlined bits are those applied to the first 2 stages.

$$\begin{array}{rll} 0 & \overbrace{00\ldots\underline{00}}^{n} & \\ 2^n-1 & 11\ldots\underline{11} & \\ 2(2^n-1) & 111\ldots\underline{10} & \\ 3(2^n-1) & 11\ldots\underline{01} & = (2^n-1) + 2(2^n-1) \end{array}$$

Each operand will eventually produce these codewords independently so that all 16 input combinations will be applied. The cycling property will then insure that all stages will be checked for carry and sum errors. For adders with a more parallel structure this analysis must be repeated.

For residue codes the problem is much simpler. The adder and the checker are completely independent units and both will receive all possible input combinations and are therefore totally self-checked.

6. CHECKING ARITHMETIC CODES

In this section, we show that, under certain conditions, essentially all known methods for checking arithmetic can be extended easily so that they are totally self-checking. First, a look at parity codes:

In the case of parity checking, the main constraints are imposed on the adder itself. The parity P_S of the sum (S = A + B) can be computed as follow:

$P_S = P_A + P_B + P_C$ where P_C is the parity of the carries.

However, in practice a fault may affect both the carry and the sum digits, producing an undetected error. One approach to the problem is to duplicate the carries i.e. generate independently pairs of carries. One set of carries is then used to compute the sum and the other set is used for checking. The adder must also have a small (= 1) or odd circuit distance so that it is fault-secure. The parity of the sum is then predicted by computing the modulo 2 sum of the parity of the operands and of the carries. The result of the sum (or subtraction) along with its predicted carry are then fed into a totally self-checking parity checker (Carter and Schneider [3]), Anderson [1]). It is possible, although more expensive, to merge prediction and checking into the double tree of exclusive-ORs forming the TSC checker.

We now turn our attention to arithmetic codes of the form $(2^n-1)N$ or residue (2^n-1). Consider Figures 1 and 2 describing serial and tree structured checking. In the first instance checking is accomplished in a serial fashion: bytes of length n are taken one at the time and accumulated modulo 2^n-1 (i.e. modulo the residue or the generator). The overall delay is, of course, the number of bytes times the assimilation time in the checker. In the lower figures , modulo 2^n-1 adders are connected in a tree fashion to form the checker. In that case the total delay is $\lceil \log_2(\text{number of bytes}) \rceil$ times the delay at each level. However the number of modulo adders is greatly increased to (# of bytes-1).

Cost and time are not the only factors involved in comparing the two approaches as their behaviours under modeled faults are quite different. In the byte serial mode it should be obvious that any single stuck-at fault in the checker will not be detected if and only if the number of bytes in the whole data word times the arithmetic circuit distance is an integer multiple of the modulus. This is the only constraint to be met in order that the fault-secureness property be satisfied. On the other hand if a tree-structured checker is used then the only requirements for it to be totally self-checking are that each module be irredundant and have maximum arithmetic circuit distance not exceeding the code capabilities.

Another point that should be brought up at this time is the distinction between AN and residue codes as far as checking is concerned. They both use the same checking structure but residue codes are separate whereas AN codes are not. The ultimate result of self-checking is to provide a coded output that determines the validity of the data. The

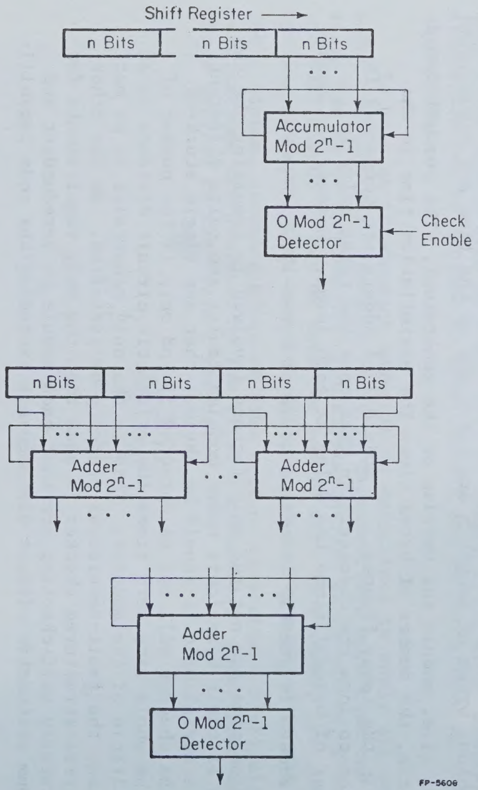

Shift Register
n Bits
n Bits
n Bits
Accumulator Mod 2^n-1
0 Mod 2^n-1 Detector
Check Enable
n Bits
n Bits
n Bits
n Bits
Adder Mod 2^n-1
Adder Mod 2^n-1
Adder Mod 2^n-1
0 Mod 2^n-1 Detector
FP-5606

Figure 1: Serial and Tree Checkers for AN Codes

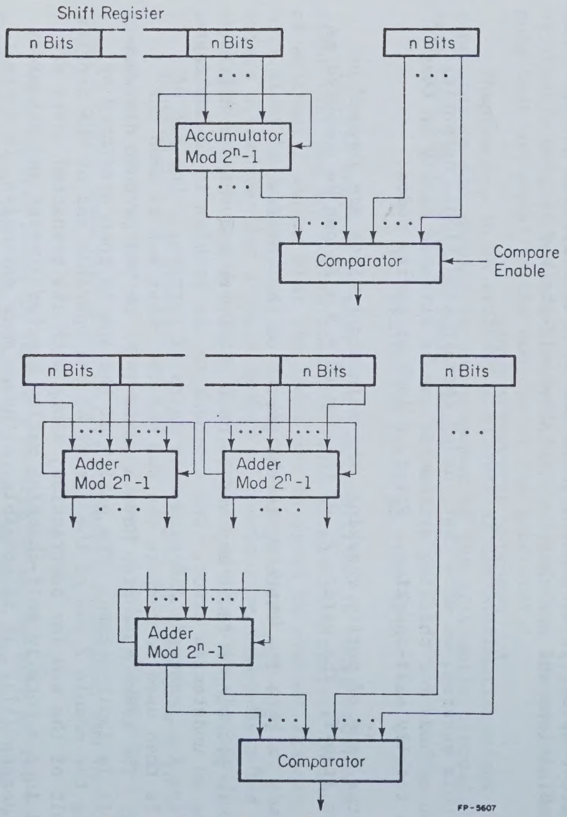

Shift Register
n Bits
n Bits
n Bits
Accumulator Mod 2^n-1
Comparator
Compare Enable
n Bits
n Bits
n Bits
Adder Mod 2^n-1
Adder Mod 2^n-1
Adder Mod 2^n-1
Comparator
FP-5607

Figure 2: Serial and Tree Checkers for Residue Codes

difference arises one time period before (in the serial mode) or one level above (in the tree approach) the final assimilation that provides the coded checker output. In the case of residue coding, the residue of the data part of a word following an operation should equal the check bits. Inasmuch AN codes are concerned the checker output is normally zero: it follows that the last two bytes to be added in the checker are complementary (1's complement) or all zeros. The all-one vector may normally never be produced in the hardware even though it is correct. Also no matter how 0 mod 2^n-1 is encoded, the checkers described before will produce the all-zero vector if and only if they are presented with all-zero inputs. From the previous considerations it should be clear that due to dependence at the final level, the checker may not operate in a fault-secure mode at that instant and may not be completely tested. In the case of serial checking the adder in the final assimilation period is neither operating in a fault-secure nor self-testing mode. However it may not matter since the probability that conditions propitious to the generation of an erroneous code output at that time are virtually nil. It may however be unsatisfactory from a rigorous point of view on self-checking. For tree-checking the problem arises also but for both AN and residuce codes where only equal or complementary inputs plus the all-zero input are available. The problem is now more acute since the final module is never fault-secure and is not completely self-tested.

The problem can be solved very easily as follows: For AN codes the inputs $X_1X_2...X_n\overline{X}_1\overline{X}_2...\overline{X}_n$ (and perhaps 00..000...0) are simply fed into a two-rail comparator with inputs $X_1\overline{X}_1X_2\overline{X}_2...X_n\overline{X}_n$ [1,3]. The 01 and 10 comparator's outputs will indicate a correct result. For residue codes simply invert one of the two bytes present before final assimilation, i.e. $X_1X_2...X_nX_1X_2...X_n$ and interleave the bits $X_1X_2..X_n\overline{X}_1\overline{X}_2..\overline{X}_n$ as before and and use a two-rail comparator. For AN codes this final piece of circuitry is totally self-checked with respect to unidirectional faults. Because of the inversion needed the logic for the final assimilation in the case of residue codes is TSC with respect to single faults only. In both cases in the serial mode additional logic (timing etc.) is needed. For tree checking the two-rail comparator and inverters (if needed) can simply be hooked up to the previous adders.

Another approach would be to design from scratch checkers with the desired properties. So far no general method that eclipses the simplicity and structure of checkers constructed out of modulo 2^n-1 adders, has been found.

There are other problems that must be looked at when designing TSC arithmetic checkers. For example special care must be exercised in two cases:

1) Design of checkers for minimally redundant arithmetic codes.
2) Design of checkers for arithmetic circuits with large amount of look-ahead or parallelism (i.e. circuits with large arithmetic circuit distance).

Clearly most designs will fall between these 2 extremes, so that guidelines that take into account each of these will always apply, but to varying extent. In the case of minimally redundant arithmetic codes, i.e. the residue 3 and 3N codes, precautions must be taken in designing both the adder and the checker so that no error will produce output changes equal to the modulus. This is the case when any two adjacent output bits are erroneous unidirectionally. Incidentally "standard" checkers for these codes are totally self-checking with just 2 outputs [4]. Fortunately for codes with identical arithmetic distance but with more redundancy this problem does not exist: any changes in boundary bits of different bytes will never cause a change equal to the modulus. However, carry propagation schemes other than the basic ripple through carry might possess just the right defeating arithmetic circuit distance in the carry propagation circuit. So as in the case of minimally redundant codes care must be taken so that circuits with large arithmetic distance do not exceed the code capabilities.

7. CONCLUDING REMARKS

It has been seen that provided some guidelines are followed conventional structures to check arithmetic codes are TSC. These guidelines stem from the study of distance and its relation to arithmetic distance.

Arithmetic codes have been seen to lend themselves naturally to the design of TSC arithmetic circuits. As observed in the discussion on distance their error coverage is best expressed in terms of arithmetic distance and when used elsewhere in a computer these codes have an error coverage which is difficult to define accurately in terms of the dominant fault model. In other words arithmetic codes generally perform well only in arithmetic circuits and other codes should be used in the non-arithmetic sections of a computer.

REFERENCES

[1] Anderson, D. A., "Design of Self-Checking Digital Networks Using Coding Techniques," Coordinated Science Laboratory Report R-527, University of Illinois, Sept. 1971.

[2] Avizienis, A., "Arithmetic Algorithms for Error-Coded Operands," Digest of the 1972 International Symposium on Fault-Tolerant Computing, Pasadena, December 1971.

[3] Carter, W. C. and Schneider, P. R., "Design of Dynamically Checked Computers," IFIP 68, Vol. 2, Edinburgh, Scotland, pp. 878-883, August 1968.

[4] Dussault, J. and Metze, G., "A Low-Cost Totally Self-Checking Checker for 3N and Residue 3 Codes," Proc. of the 14th Allerton Conference, Monticello, Sept. 30-Oct. 1, 1976.

[5] Ho, D. A., "The Design of Totally Self-Checking Systems," Coordinated Science Laboratory Report R-723, University of Illinois, April 1976.

[6] Kolupaev, S. G., "Self Testing Residue Trees," Technical Report No. 49, Digital Systems Laboratory, Stanford University, August 1973.

[7] Massey, J. L. and Garcia, O. M., "Error Correcting Codes in Computer Arithmetic," Advances in Information Sciences, Chap. 5.

ACKNOWLEDGMENT

The work reported in this paper was supported mainly by the Department of Education of the Province of Québec and in part by the Joint Services Electronics Program (U.S. Army, U.S. Navy, and U.S. Air Force) under Contract No. DAAB-07-72-C-0259.

EASILY TESTABLE BINARY CIRCUITS CONSTRUCTED WITH MULTIVALUED MORPHIC GATES

JEAN DUSSAULT
Coordinated Science Laboratory, Urbana, Illinois

The idea of embedding binary circuits into multivalued circuits to enhance testability has been proposed by Vranesic [1]. He uses the 3-valued Post algebra and the resulting ciruits are testable using a very small number of tests. This short paper introduces a method whereby any circuit (combinational or sequential)constructed out of morphic gates to be described is completely tested for any multiple stuck-at faults using exactly 2 tests. In our case the morphic gates are four-valued and the binary function is embedded as follow:

		y			
	f(x,y)	a	0	1	b
x	a	a	*	*	*
	0	*	binary		*
	1	*	gate		*
	b	*	*	*	b

* = 0 or 1

Morphic gate of a
2-input binary gate

x	x
a	a
0	1
1	0
b	b

Morphic inverter

Clearly the two tests are the all-a and all-b vectors.Any gate input or output stuck at a value other than the value of the components of the test vector being applied produces a 0 or a 1 which is then propagated to the output. More precisely the all-a vector tests for lines stuck at 0,1 and b whereas the all-b vector tests for lines stuck at 0,1 and a. Reconvergent fanout as well as masking in general do not constitute a problem as faults tested in that way are not maskable. Although a and b appear to be ordered in a specific manner with respect to 0 and 1 i.e. a<0<1<b, this particular ordering need not be preserved in practical implementation and only the functional mappings have to be consistent with the table.

Indeed the viability of this idea resides in the existence of a suitable electronic implementation. Of course only one type of functionally complete morphic gate needs to be implemented. Larger blocks (universal logic modules, for example) with such properties may be envisaged. Further work along those lines is currently being done.

REFERENCE

[1] Vranesic Z.G., "Multivalued Circuits in Fault Detection of Binary Logic Circuits," Proceedings of the 1976 Canadian SRE Reliability Symposium, June 1976, Ottawa, pp.25-33.

This work was supported mainly by the Department of Education of the Province of Québec and in part by the Joint Services (U.S. Army, U.S. Air Force, U.S. Navy) Electronics Program under grant no. DAAB-07-72-C-0259.

INDEX OF AUTHORS

Agarwal, V. K. 732
Allemong, J. J. 476
Baras, J. S. 68
Barmish, B. R. 347
Basu, S. 221
Bedi, J. S. 452
Bentley, J. L. 193
Bixby, R. E. 547
Blankenship, G. L. 671
Brock, H. W. 613
Brockett, R. W. 671
Cariolaro, G. L. 244
Castanon, D. A. 595
Chammas, J. 451
Chen, G. 496
Chow, J. H. 476
Church, R. W. 163
Chwa, K. Y. 577
Chyung, D. H. 94
Cohn, D. L. 254
Conan, J. 451
Coraor, L. D. 162
Corsini, P. 337
Costello, D. J., Jr. 439, 449
D'Alessandro, J. 640
D'Andrea, N. A. 289
Davis, A. M. 632
Davison, E. J. 298
DeMillo, R. 186
de Pedro, H. 450
Devroye, L. P. 211
Dobkin, D. 186
Dussault, J. 751, 759
Elrazaz, Z. 305
El-Sherief, H. 231
Erlandson, R. F. 460
Gabow, H. N. 557
Gallagher, N. C. 264
Gardner, B. F., Jr. 96
Gardner, B. W. 700
Gault, J. W. 120
Giordano, A. 450
Giuli, D. 237
Gopalsami, N. 486
Groumpos, P. P. 314
Haccoun, D. 451
Hajek, B. 406
Hakimi, S. L. 577
Hazony, D. 623, 642
Hazony, Y. 623, 642
Hickin, J. 104
Ibarra, O. H. 202
Johnson, D. B. 1
Kaliski, M. E. 161
Kamal, A. K. 452
Kassam, S. A. 517
Kazakos, D. 415, 438
Kazakos, P. P. 415, 503
Kim, C. E. 379
Klein, Q. L. 161
Kleitman, D. J. 165
Krishnaprasad, P. S. 661
Kuhl, J. G. 146
Kumar, P. R. 680
Kung, H. T. 34
Leitman, G. 604
Li, R. 415
Lipton, R. J. 186
LoCicero, J. L. 271
Lodi, E. 389
Luccio, F. 8, 389
Manacher, G. K. 390
Marcus, S. I. 52, 652
Masson, G. M. 732

Meditch, J. S. 690
Mengali, U. 280
Mese, E. D. 237
Metze, G. 136, 751
Meyer, A. R. 165
Meyer, G. G. L. 727
Mitter, S. K. 62
Monopoli, R. V. 78
Morefield, C. L. 700
Moustakas, E. A. 632
Mulder, L. Th. N. M. 254
Murata, T. 163
Nair, R. 136
Nishizeki, T. 567
Ogawa, H. 567
Pagli, L. 8
Papadimitriou, C. H. 368
Papanicolaou, G. C. 44
Payne, H. J. 700
Perry, R. 640
Peterson, R. 439
Pezzani, E. 280
Picchi, G. 337
Pippenger, N. 25
Plackovic, R. 324
Poor, H. V. 507
Prati, G. 337
Prezas, D. P. 271
Price, E. L. 429
Quam, J. G. 718
Raju, D. S. 448
Reddy, S. M. 146, 718
Reisenfeld, S. 537
Rivest, R. L. 165
Robinson, J. P. 162, 448
Rouse, W. B. 466
Russo, F. 289
Sahni, S. 175
Saito, N. 567
Sanathanan, C. K. 486
Sankaran, J. 347
Savage, J. E. 359
Schkolnick, M. 398
Schmitendorf, W. E. 587
Scott, P. D. 314
Shamos, M. I. 193
Shin, J. G. 517
Singh, H. 452
Sinha, A. S. C. 114
Sinha, N. K. 104, 231, 305
Snyder, D. L. 156
Spencer, J. 165
Sripad, A. B. 156
Swamy, S. 359
Thomas, J. B. 507, 527
Thompson, C. D. 398
Towsley, D. 709
Tripathi, N. K. 298
Vagliani, F. C. 244
VandeLinde, V. D. 221, 429
Van Schuppen, J. H. 680
Wan, H. Y., Jr. 604
Wang, D. L. 547
Wise, G. L. 211, 652
Wojcik, A. S. 129
Wolfe, J. K. 709
Wolfe, C. A. 700
Wong, C. K. 16
Wood, M. D. 527
Woodard, S. E. 742
Yao, A. C. 553
Yao, K. 421, 537
Young, K. D. 85
Young, S. K. 62
Yu, C. C. 449

UNIVERSITY OF ILLINOIS-URBANA
621.3811AL5 C001
PROCEEDINGS / ALLERTON CONFERENCE ON COM
1977
3 0112 008149129